Filters

John Carwardine

Outline

- Application of filters to beam stability
- Ideal frequency-selective filter characteristics
- Characteristics of common analog filters
- Anti-alias filters
- Averaging as a filter
- FIR digital filters
- IIR digital filters

APPLICATIONS OF FILTERS TO BEAM STABILITY

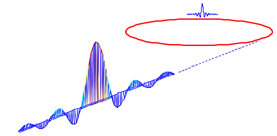
Applications of filters to beam stability

- Anti-alias filters prior to A/D conversion
  - Stringent requirements not to contaminate signals as they are digitized.

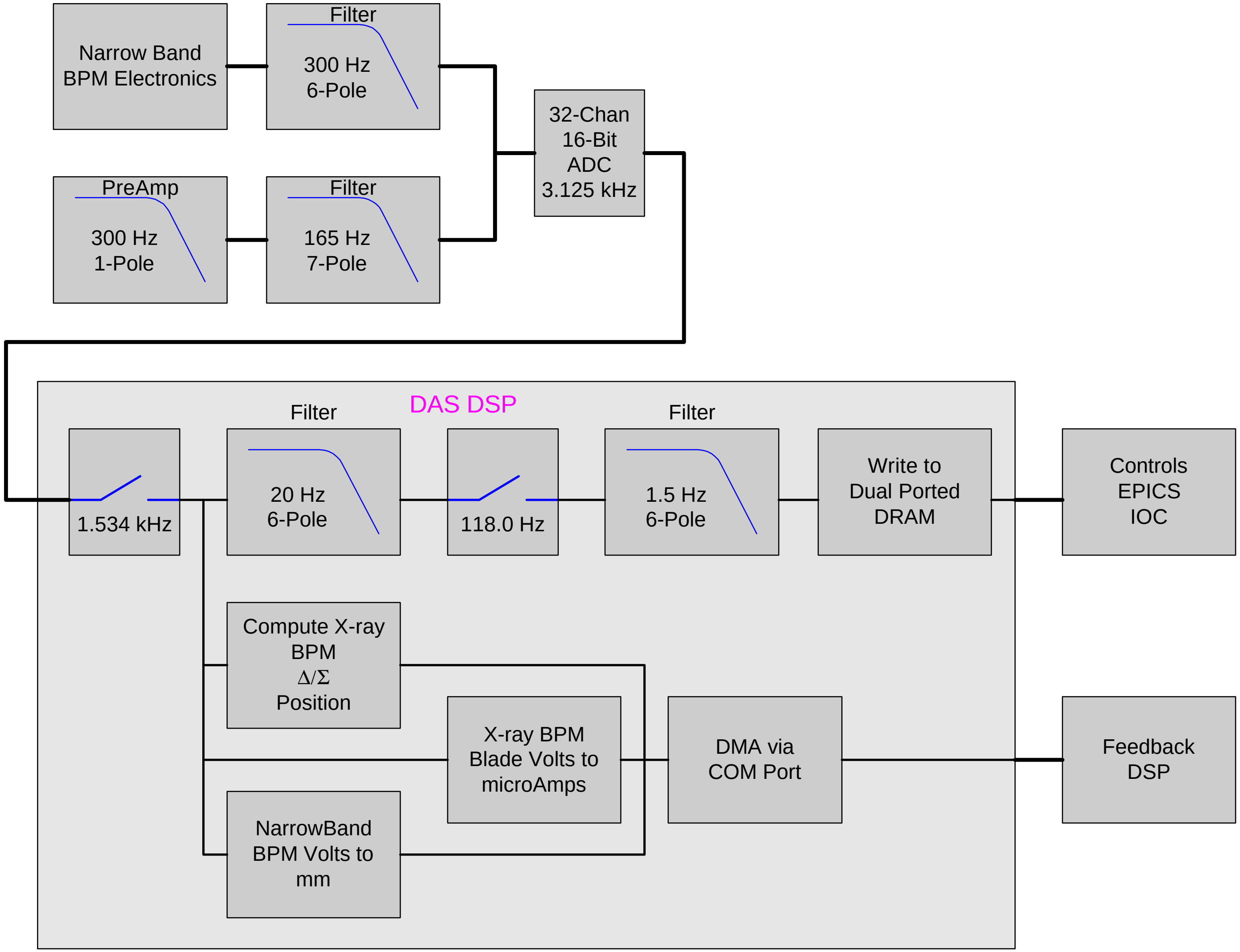
- Anti-alias filters prior to sample-rate conversion (down-sampling)
  - Similar stringent requirements to anti-alias filters for A/D conversion.

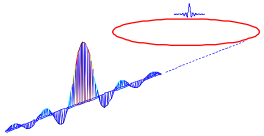
- Implementation of digital regulator functions.

- Implementation of signal processing algorithms (eg measurement of beam motion within specified frequency bands).



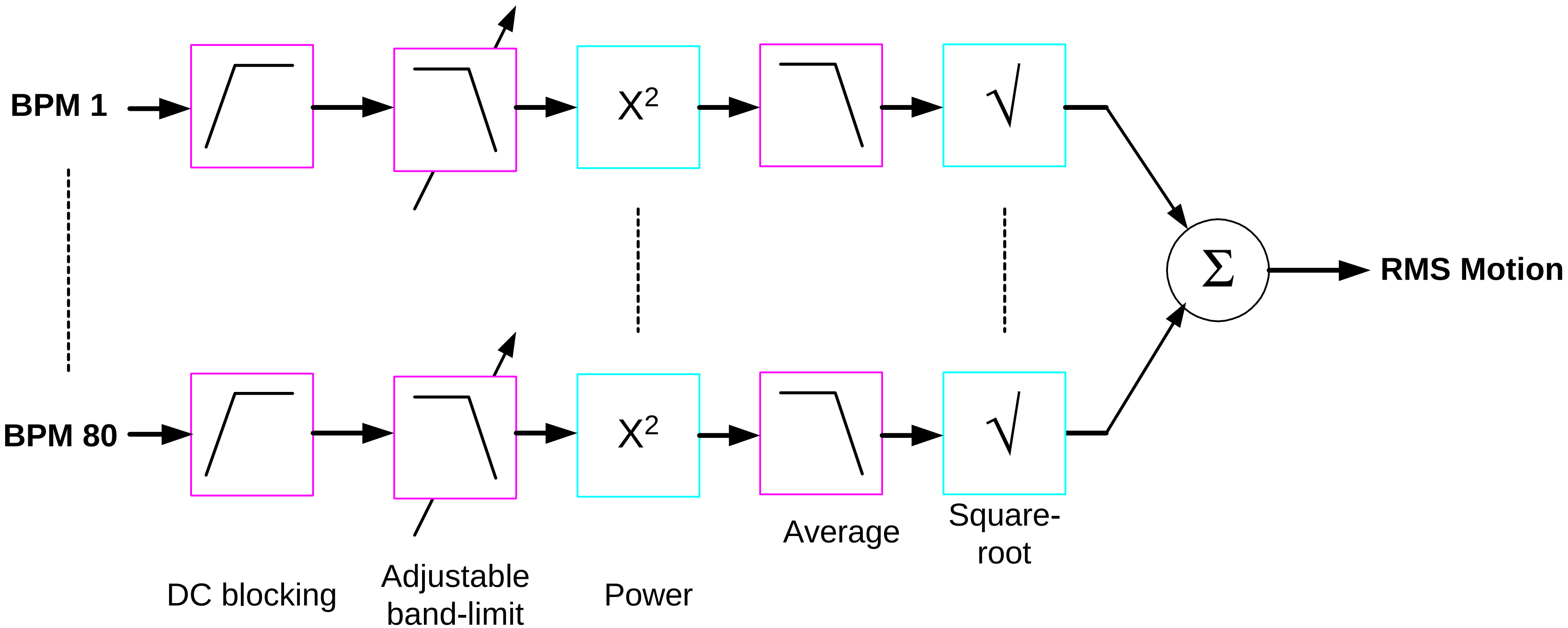
Analog front-end of APS x-ray and Narrowband rf bpms





Real-time RMS orbit motion calculations

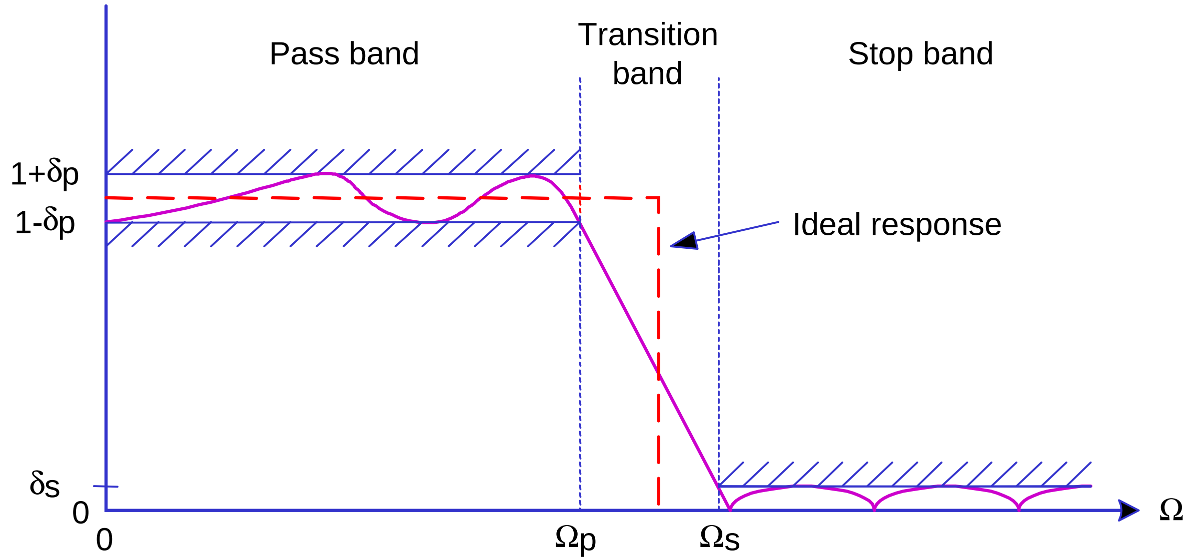
- New real-time measurement of the APS orbit motion



PROPERTIES OF COMMON ANALOG FILTERS

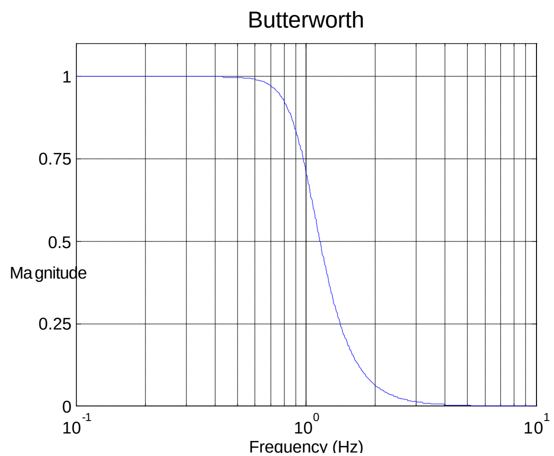
Frequency Response of Practical Filters

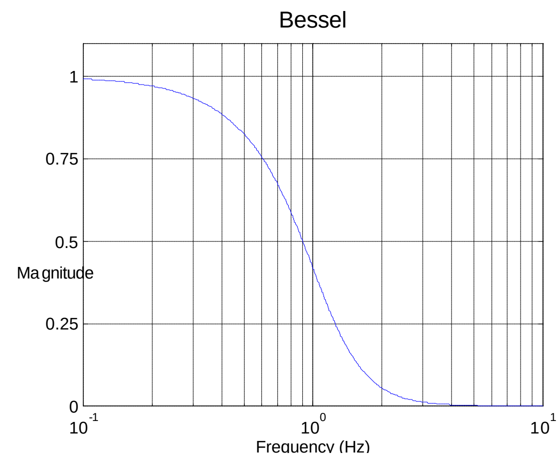
- When a realizable impulse response is generated, the frequency response of the resulting filter is compromised from the ideal response
  - The passband may not be flat
  - There is a finite width to the transition from passband to stopband
  - The stopband will not have infinite attenuation
  - The phase response will not be zero for all frequencies.

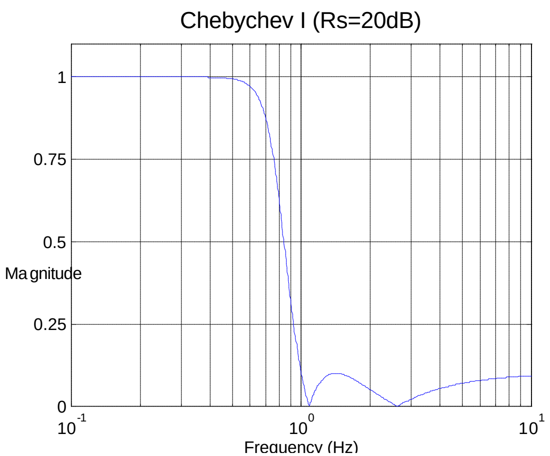


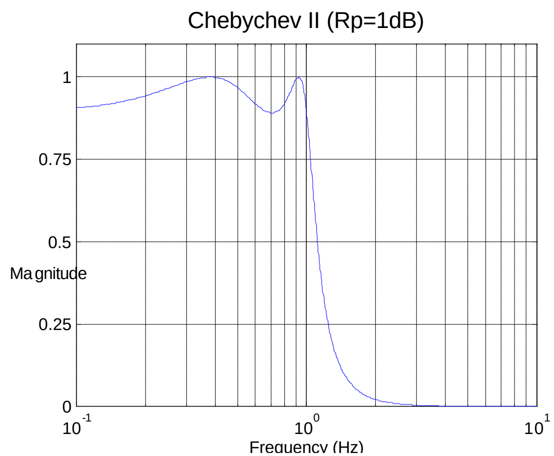
Magnitude Response of Common Analog Filter Types

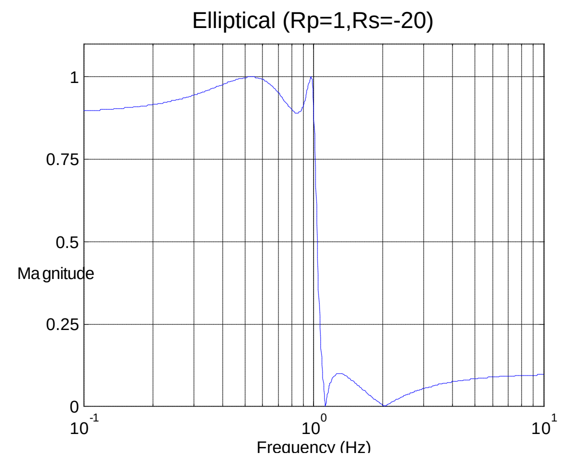
- The following are all 4th-order analog lowpass filters with cutoff at 1Hz











Basic Properties of Common Analog Filter Types

	Passband	Stopband	Key benefits
Butterworth	Flattest	-20N dB/decade	Maximally flat in passband
Chebyshev Type I	Equiripple	-20N dB/decade	Faster initial roll-off than Butterworth
Chebyshev Type II	Flat	Equiripple	Faster roll-off than Butterworth
Elliptic	Equiripple	Equiripple	Narrowest transition band
Bessel	Monotonic	-20N dB/decade	Linear-phase in passband

Specifying Analog Filters with $|H_a(jw)|^2$

- Consider the following Laplace transfer function

$$H_a(s) = \frac{1}{a \cdot s^2 + b \cdot s + c}$$

- The magnitude response is computed by setting $s = jw$ and computing the magnitude of the resulting expression

$$\left|H_a(j\Omega)\right| = \left|\frac{1}{a \cdot (j\Omega)^2 + b \cdot (j\Omega) + c}\right|$$

- However, the magnitude response can also be computed from the following product

$$\begin{aligned} |H_a(jw)|^2 &= H_a(jw) \cdot H_a(-jw) \\ &= \frac{1}{a(j\Omega)^2 + b(j\Omega) + c} \cdot \frac{1}{a(-j\Omega)^2 + b(-j\Omega) + c} \\ &= \frac{1}{a^2\Omega^4 + (b^2 + 2ac)\Omega^2 + c^2} \end{aligned}$$

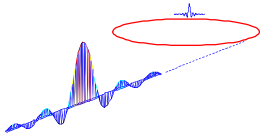
- The magnitude-squared of any Laplace transfer function can be computed from this product which always results in a rational polynomial of powers of w^2.

Transfer Functions of Lowpass Analog Filters

- Commonly, the transfer functions of analog lowpass filters are of the form

$$\left|H_a(s)\right|^2 = \frac{1}{1+P_N(s)^2}$$

- Where $P_N(s)$ is a polynomial of order N in s the form of which depends on the chosen filter type.
- Examples for $P(s)$ are:
  - Butterworth filters have $P_N(s) = s^N$
  - Chebyshev and Elliptical filters use Chebyshev polynomials of order N.

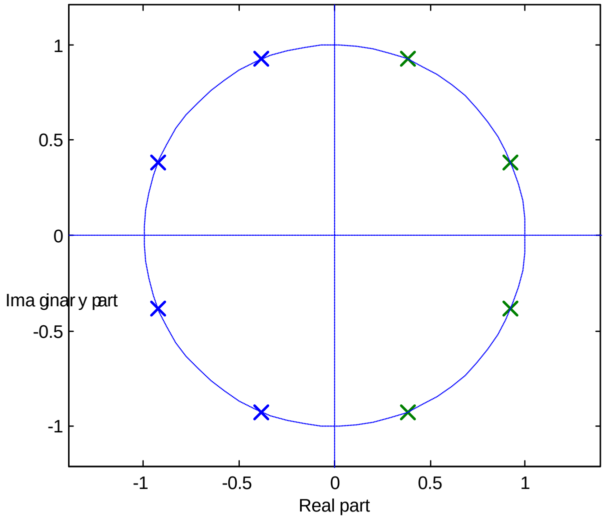


Butterworth Filters (cont)

- The magnitude-squared response of an N^{th} order Butterworth filter is given by

$$|H_a(j\Omega)|^2 = \frac{1}{1+(\Omega/\Omega_c)^{2N}}$$

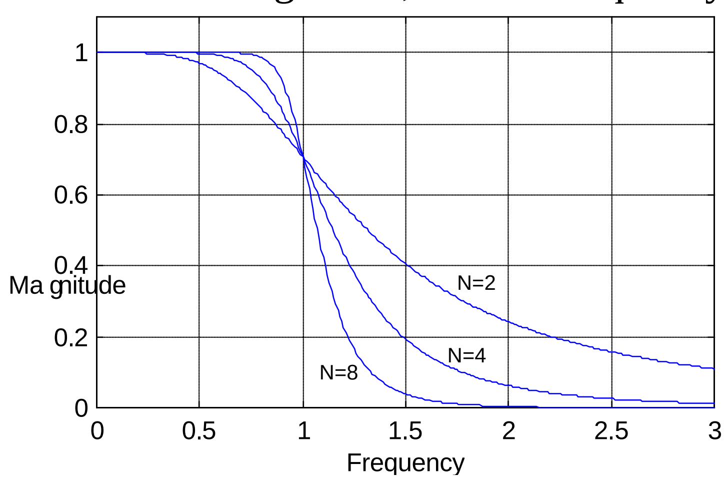
- The poles of the Butterworth magnitude-squared response all lie on a circle of unit radius in Laplace-space.



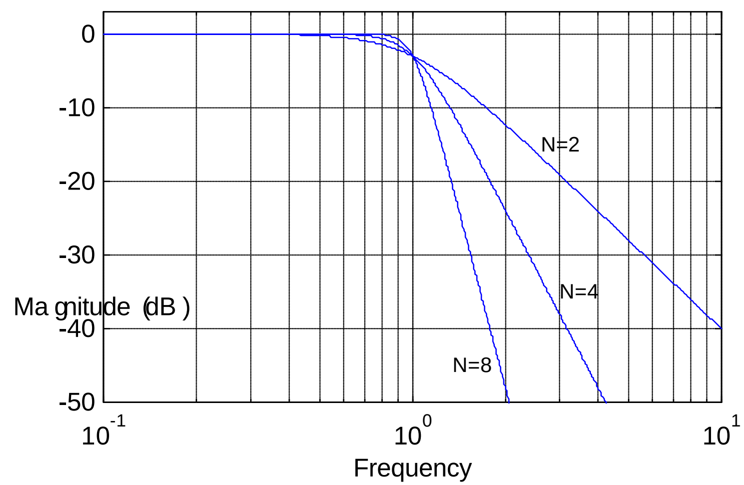
Butterworth Filters

- Butterworth filters are maximally-flat.
- There is no ripple in either the passband or stopband.
- The magnitude-response of an N^{th}-order filter rolls off at 20*N* dB/decade.
- The stopband phase delay of an N^{th}-order filter is -90N degrees.
- A Butterworth filter can be completely described by its -3dB cutoff frequency *Wc*, and its order *N.*

Linear Magnitude, Linear Frequency



Magnitude in dB, Log Frequency



Transfer Functions of Normalized Butterworth Lowpass Filters

Filter Order	Coefficients for each power of s								
	S^8	S^7	S^6	S^5	S^4	S^3	S^2	S^1	S^0
1								1	1
2							1	1.4142	1
3						1	2	2	1
4					1	2.6131	3.4142	2.6131	1
5				1	3.2361	5.2361	5.2361	3.2361	1
6			1	3.8637	7.4641	9.1416	7.4641	3.8637	1
7		1	4.4940	10.0978	14.5918	14.5918	10.0978	4.4940	1
8	1	5.1258	13.1371	21.8462	25.6884	21.8462	13.1371	5.1258	1

- All these filters are normalized (ie their -3dB cutoff frequency is 1rad/s).
- For example, the 4th order Butterworth lowpass filter is described by the transfer function

$$H(s) = \frac{1}{s^4 + 2.6131s^3 + 3.4142s^2 + 2.6131s + 1}$$

Butterworth Lowpass Filter Design Example

- *Determine the lowest order of a Butterworth filter that has a -3dB cutoff at 1kHz, and minimum attenuation of 40dB at 5kHz.*

Solution

- We'll use the following expression for a Butterworth filter to compute the order.

$$\frac{1}{1+(\Omega/\Omega_c)^{2N}} = |H_a(j\Omega)|^2$$

Substituting known values,

$$\frac{1}{1+(2p\cdot 5000/2000p)^{2N}} = 0.01^2$$

$$N = \frac{1}{2}\frac{\log_e(10^4-1)}{\log_e(2p\cdot 5000/2000p)}$$

$$N = 2.86 \rightarrow 3$$

The normalized 3rd-order Butterworth filter is given by

$$H(s) = \frac{1}{s^3+2s^2+2s+1} = \frac{1}{(s+1)\left(s+e^{j2p/3}\right)\left(s+e^{-j2p/3}\right)}$$

ANTI-ALIAS FILTER DESIGN

Anti-Alias Filter Considerations

- Maintain accuracy commensurate with ADC resolution
  - Reduce alias contamination below quantization noise of ADC
  - Keep filter pass-band attenuation within ADC resolution
- Parameters to adjust
  - Sample Frequency
  - Filter Type
  - Filter cutoff frequency
  - Filter Order
- Other factors
  - Filter phase shift may be important consideration in stability of feedback applications
  - Filter pass band undulations may be undesirable in high resolution measurement applications
    - No pass band undulations - Butterworth, Bessel, Chebychev I
    - Pass undulations - Chebychev II, Eliptical
  - Filter roll-off affects amplitude of frequencies near cutoff

Anti-Alias Filter Considerations

- Relation between sampling frequency and desired attenuation
- For a Butterworth Filter

$$|H(jw)|^2 = \frac{1}{1+\left(f/f_c\right)^{2N}} \qquad |H(jw)|_{dB} = 10\log_{10}\left[\frac{1}{1+\left(f/f_c\right)^{2N}}\right]$$

- Difference in dB between passband frequency f_p and any frequency f_a

$$10\log_{10}\left[\frac{1+\left(f_a/f_c\right)^{2N}}{1+\left(f_p/f_c\right)^{2N}}\right] \approx 10\log_{10}\left[\left(\frac{f_a}{f_p}\right)^{2N}\right] = 20N\log_{10}\left[\frac{f_a}{f_p}\right]$$

Mitra p317

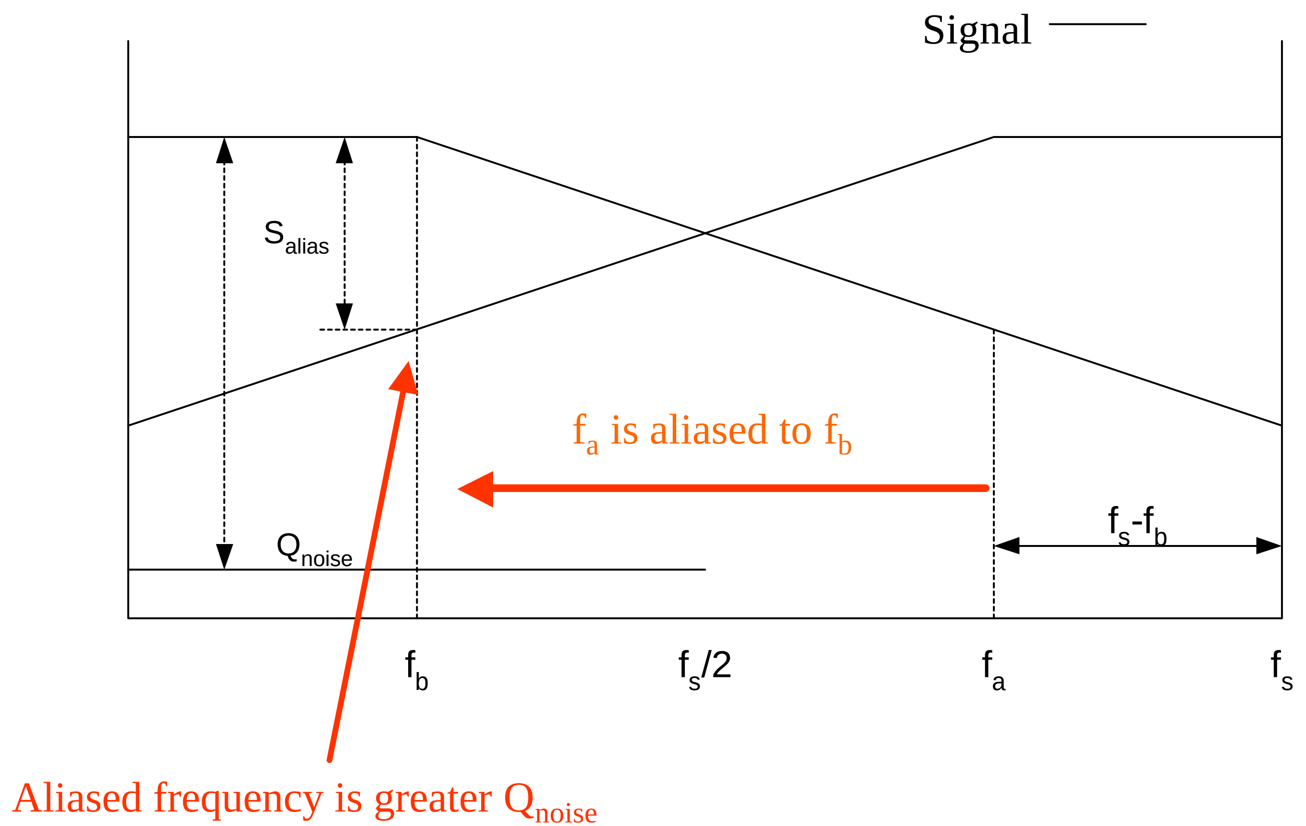
- We select the lowest aliased frequency to fall at f_p so $f_a = f_s - f_p$ therefore:

$$dB\ Attenuation \approx 20N\log_{10}\left[\frac{f_s}{f_p} - 1\right]$$

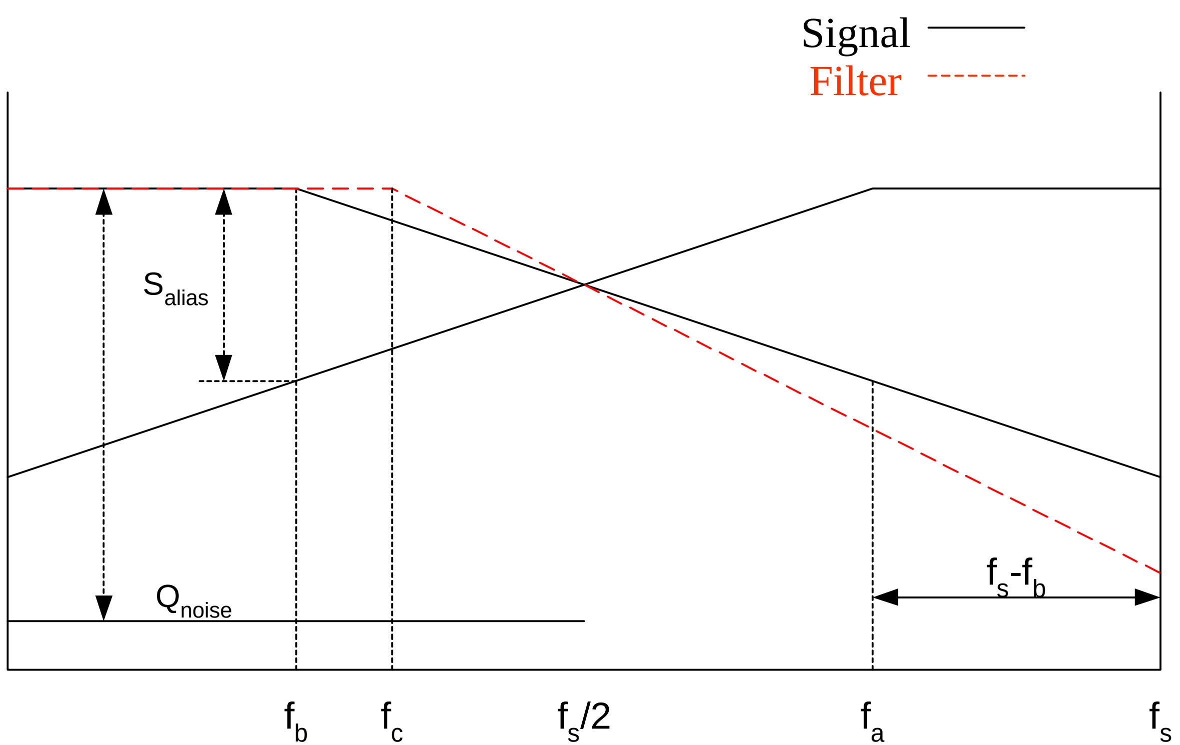
- The above equation relates desired attenuation to Butterworth Filter order and the ratio of the sampling frequency to the pass band frequency
- The following table evaluates the above expression for f_s/f_p ratios 3-10

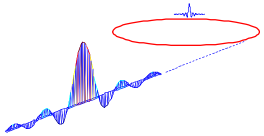
3	4	5	6	7	8	9	10
6.02N	9.54N	12.04N	13.98N	15.56N	16.9N	18.06N	20N

Anti-Alias Filter Considerations

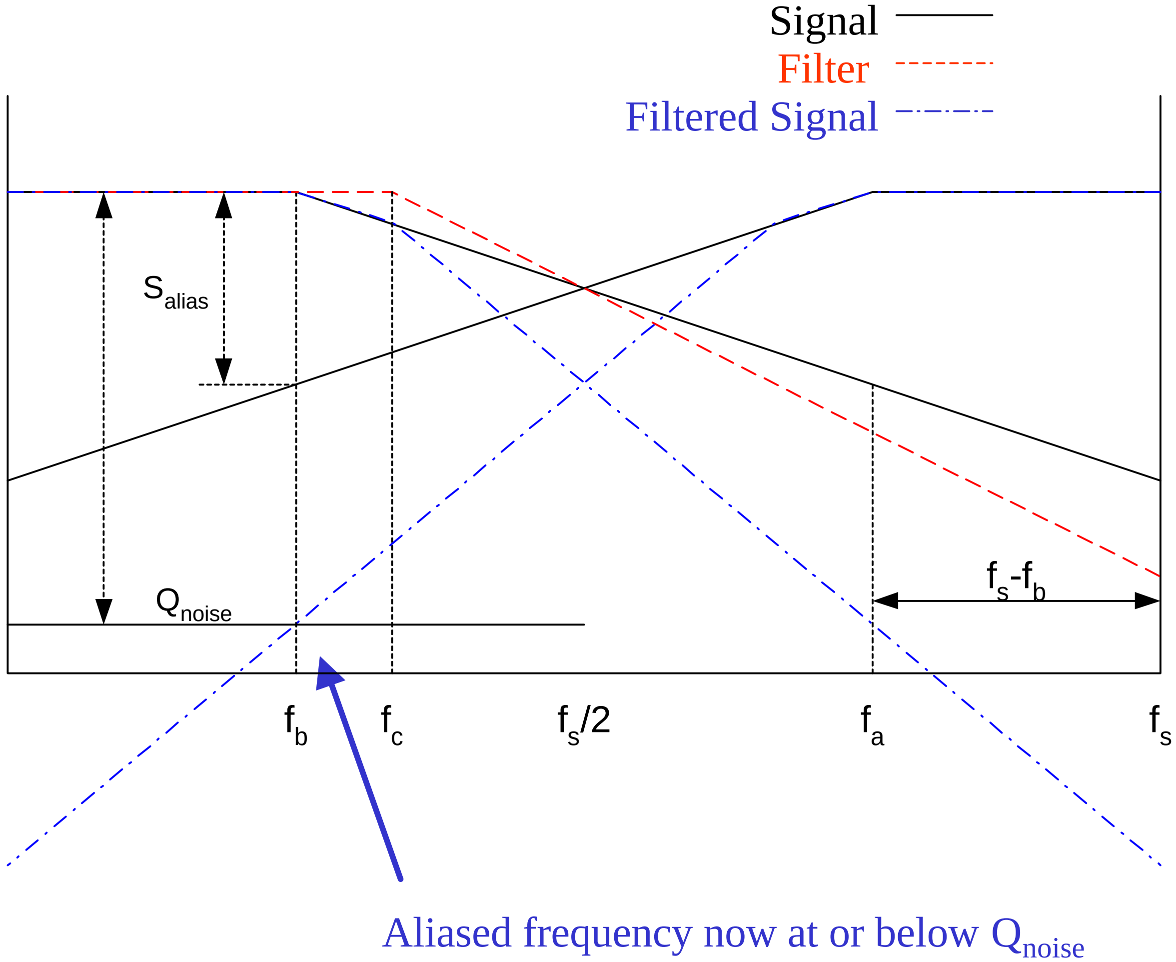


Anti-Alias Filter Considerations





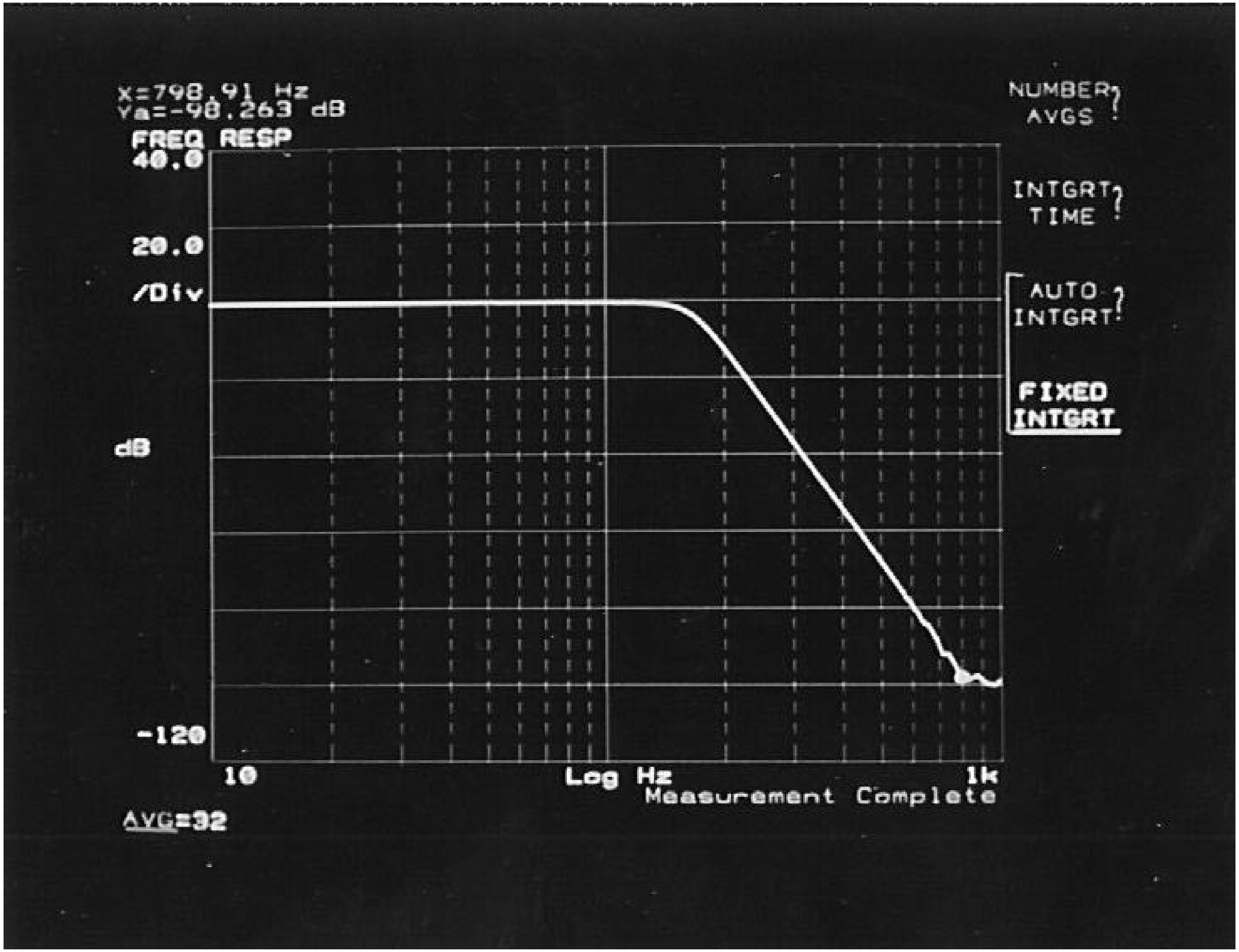
Anti-Alias Filter Considerations



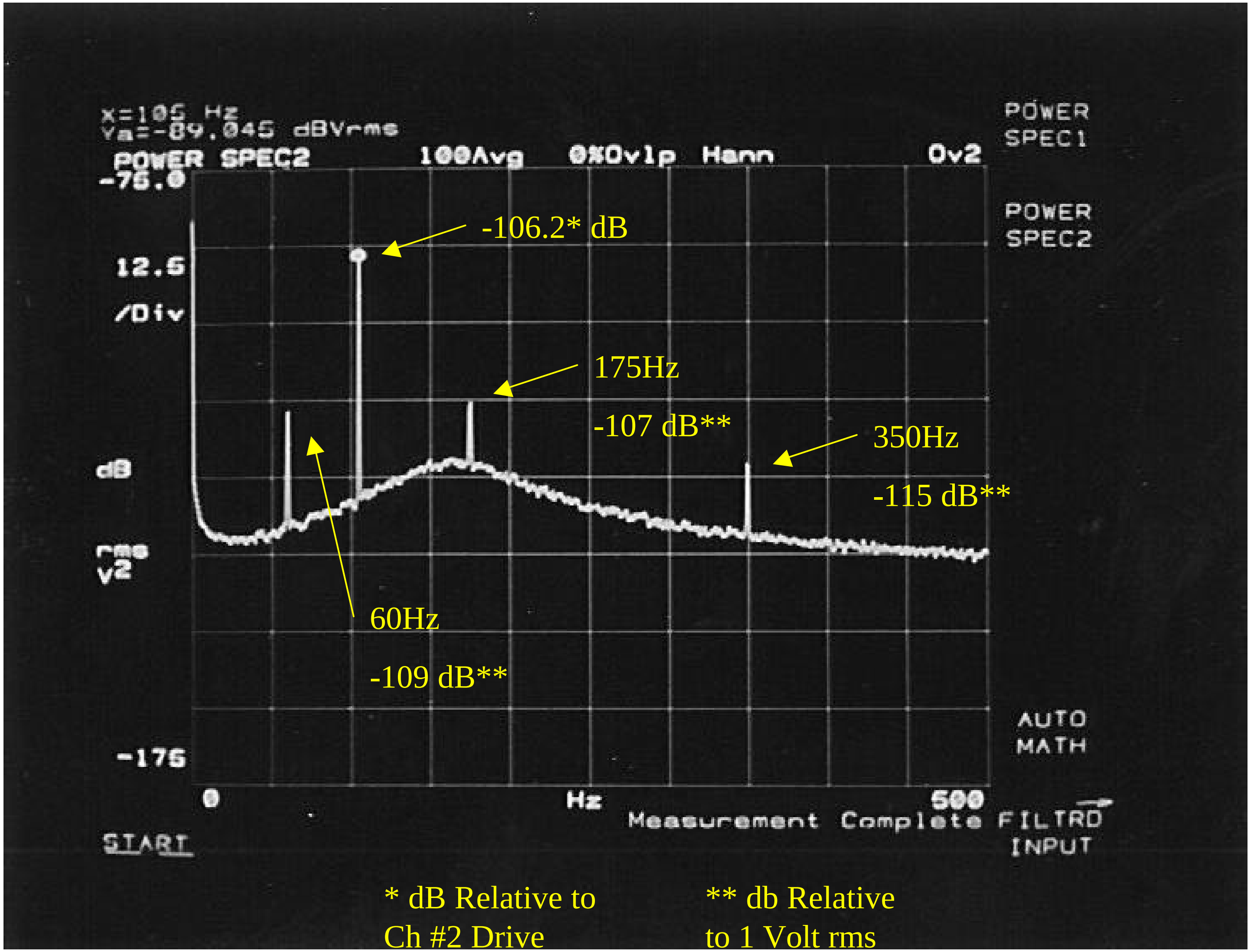
Measured Filter Performance

Bandwidth (3 dB)	165 Hz
Attenuation (at 800 Hz)	98 dB
Spurious Free Dynamic Range (45 Hz Full-Scale Input)	90 dB
Noise and Pickup	-115 dB
Adjacent Channel Crosstalk (At 105 Hz)	-116 dB

Filter Frequency Response (Average = 32)

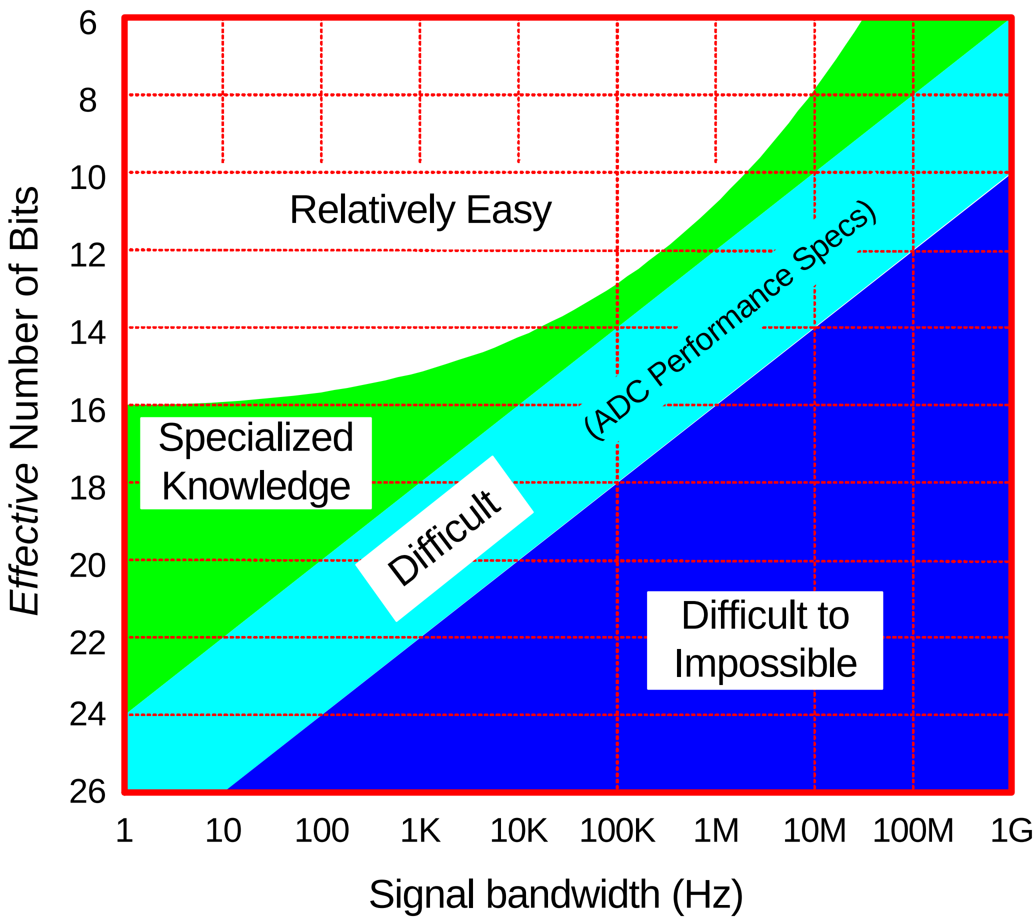


Cross Talk on Channel #1



Digitizer performance trade-offs

- Getting even 16-bit performance is not as simple as just using a 16-bit digitizer!

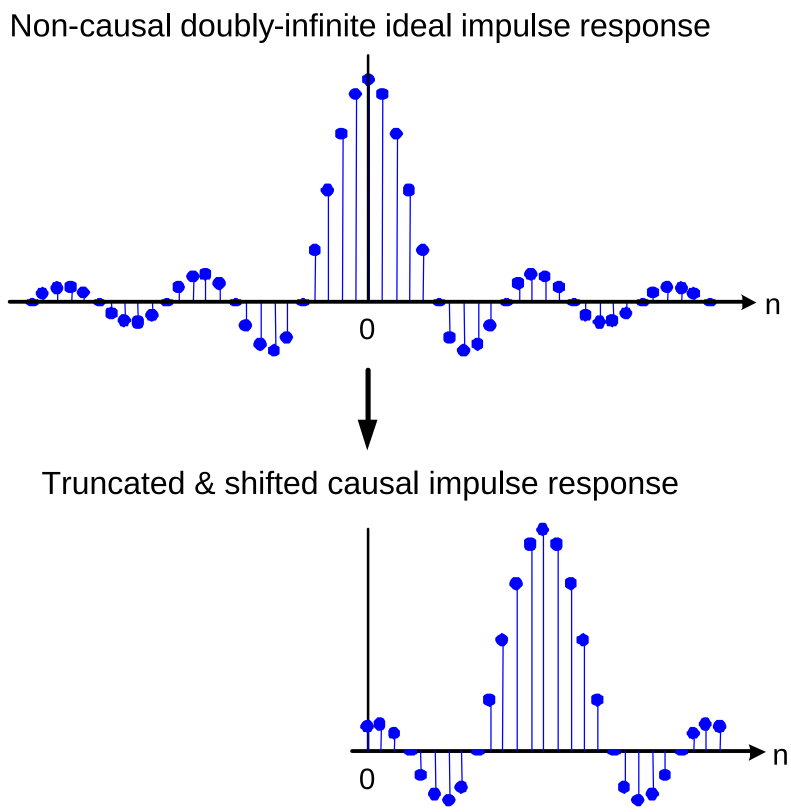


Ref: “Practical Limits of Analog-to-Digital Conversion” (Jerry Horn)

FINITE IMPULSE RESPONSE (FIR) DIGITAL FILTERS

FIR Filter Design by Impulse Response Truncation (IRT)

- In the IRT method of designing an FIR filter, we take the impulse response of the idealized impulse response, truncate it to (say) *2M+1* samples, and shift it by *M* samples to make the impulse response causal.

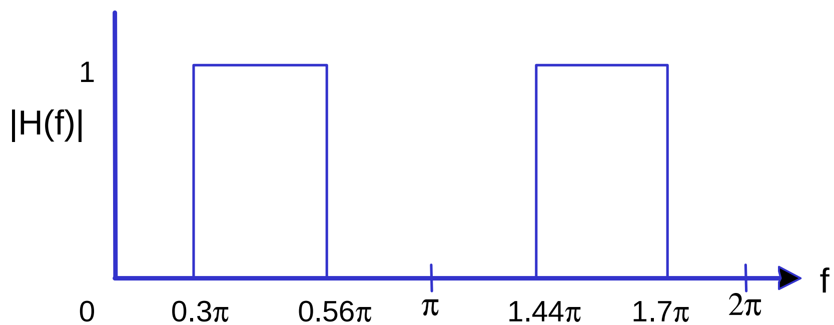


FIR Filter Design Example Using IRT

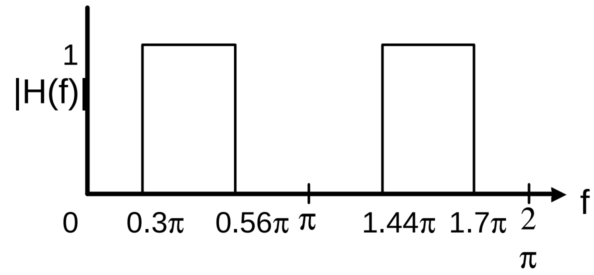
Design a bandpass filter with band edges at 0.3p and 0.56p and an impulse response of length 31.

<u>Solution</u>

- The frequency response must be specified from 0 to 2π, in order to do the inverse Fourier transform.
- The magnitude of *H(F)* will be unity from *0.3p* to *0.56p* and from *1.44p* to *1.7p* and zero elsewhere, as shown below



FIR Filter Design Example by IRT (cont)



- First, we'll compute the ideal impulse response

$$h[n] = \int_0^1 H(f)e^{jw}df$$

$$= \int_{0.3}^{0.56} e^{jw}df + \int_{1.44}^{1.7} e^{jw}df$$

$$= \frac{1}{jn}\left[e^{jwn}\right]_{0.3}^{0.56} + \frac{1}{jn}\left[e^{jwn}\right]_{1.44}^{1.7}$$

$$= \frac{1}{jn}\left[e^{j0.56n} - e^{jn0.3n} + e^{j1.7n} - e^{j1.44n}\right]$$

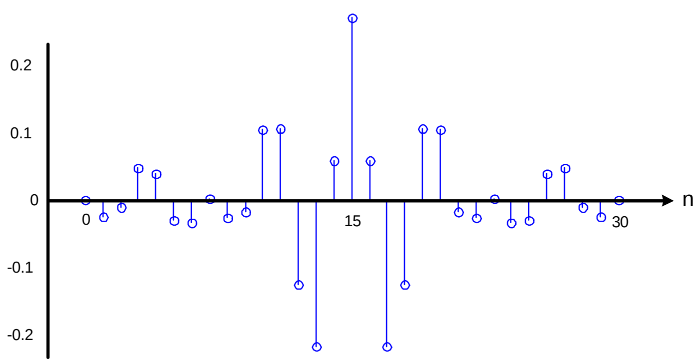
$$= \frac{1}{jn}\left[e^{j0.56n} - e^{j0.3n} + e^{-j0.3n} - e^{-j0.56n}\right]$$

$$= 0.56\frac{\sin(0.56n)}{0.56n} - 0.3\frac{\sin(0.3n)}{0.3n}$$

FIR Filter Design Example by IRT (cont)

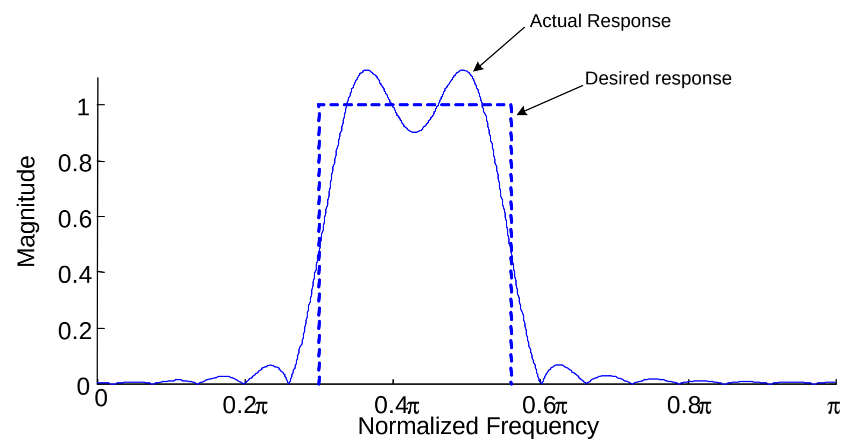
- Truncate the sequence to 31 points by defining that the sequence be zero outside the range *-15 £ n £ 15*.
- The sequence is then made causal by shifting the truncated impulse response to the right by 15 points.
- The final impulse response and the corresponding frequency response are shown below

31-point Impulse Response



$$h[n] = 0.56\frac{\sin(0.56n)}{0.56n} - 0.3\frac{\sin(0.3n)}{0.3n}$$

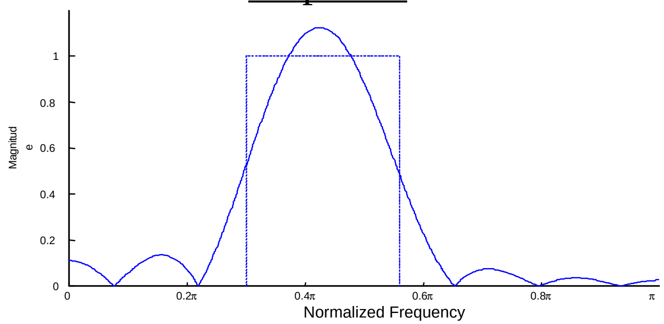
Frequency Response of 31-point Filter



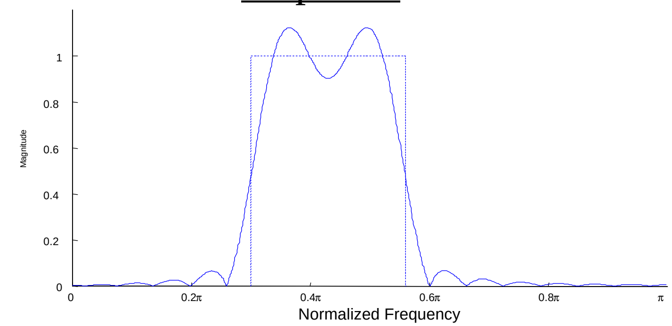
Frequency Response vs Length of Truncated Impulse Response

- More points gives a better approximation to the desired (ideal) frequency response

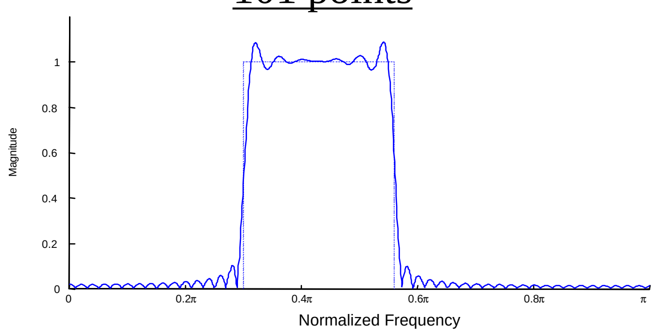
11 points



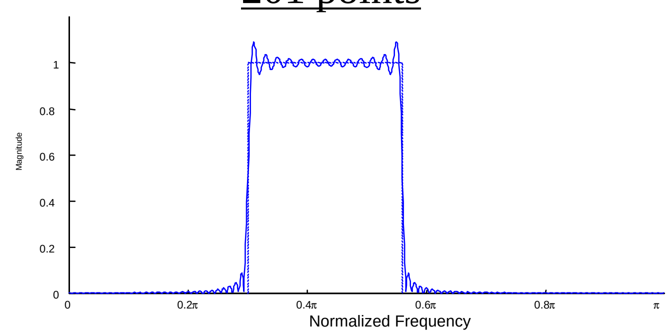
31 points



101 points



201 points



…but there is no change in the amplitude of the passband or stopband ripple.

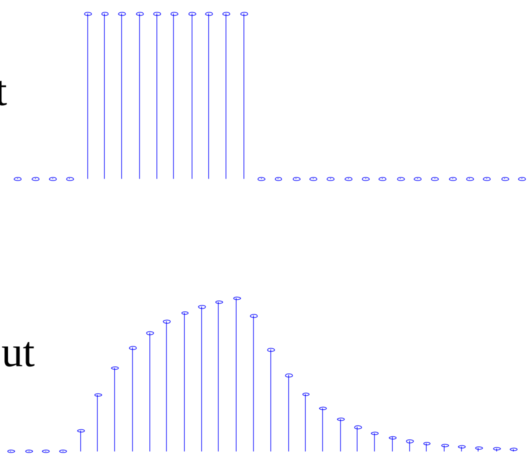
Phase Distortion and Linear Phase Response

- Nonlinear-phase filters (eg a simple IIR lowpass filter) introduce distortion because difference frequency components depart from the filter at different times.

Simple IIR Filter (a=0.75)
(Non-linear Phase)

Input

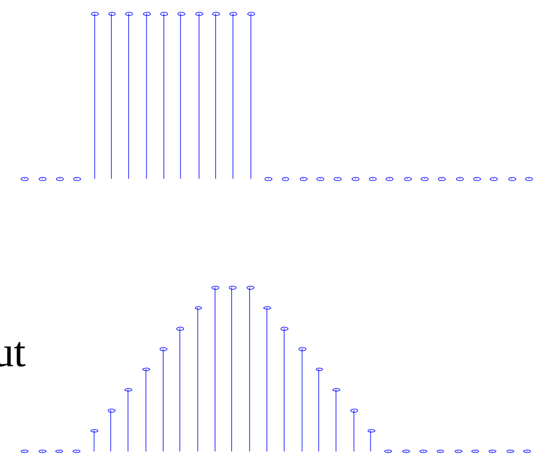
Output



8-Point FIR Averager
(Linear Phase)

Input

Output



- Whether it is better to have phase distortion or a time-delay will depend on the application (eg in feedback/control, the time-delay can significantly reduce bandwidth).

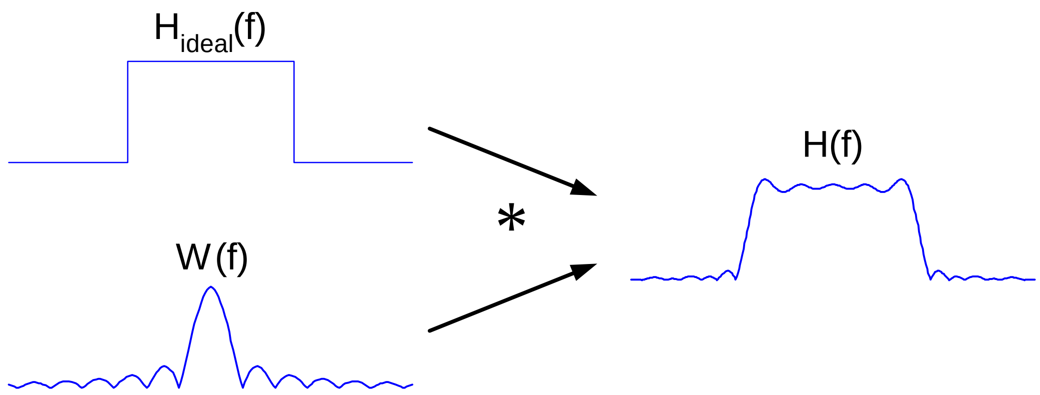
Gibbs Effect and the Impulse Response Truncation Method

- The truncation process is in effect multiplication of the ideal impulse response by a rectangular window (c.f. windowing in the DFT).

$$h[n] = h_{ideal}[n] \cdot w[n]$$

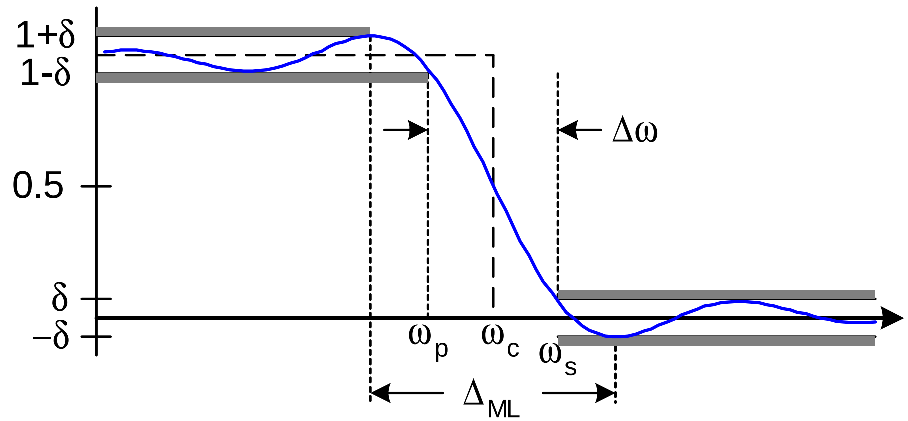
- In the frequency domain, this means the actual frequency response is the convolution of the ideal response and the frequency response of the window function

$$H[w] = H_{ideal}[w] * W[w]$$



FIR Filter Design by Windowing

- The same window functions discussed in relation to the DFT can be used in place of the rectangular window (truncation).
- Windows used for FIR filter design include *Hann, Hamming,* and *Blackman*.
- Properties of filters designed with these windows are shown below

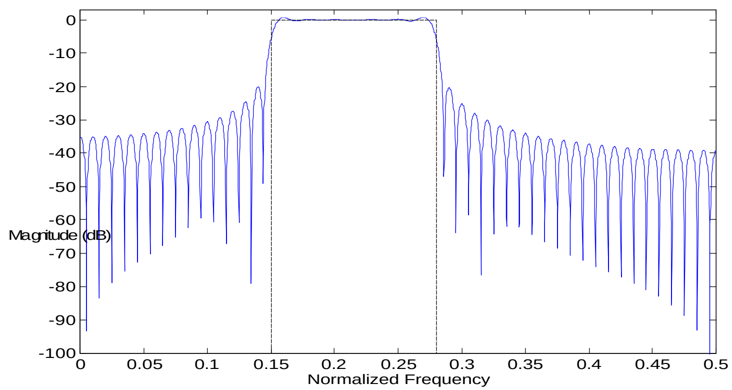


Window	Main-lobe width (D_{ML})	Transition width (Dw)	d	Passband Ripple (dB)	Stopband Ripple (dB)
Rectangular	4π/(2M+1)	0.92π/M	0.09	0.75	-21
Hanning	8π/(2M+1)	3.11π/M	0.0063	0.055	-44
Hamming	8π/(2M+1)	3.32π/M	0.0022	0.019	-53
Blackman	12π/(2M+1)	5.56π/M	0.0002	0.0017	-74

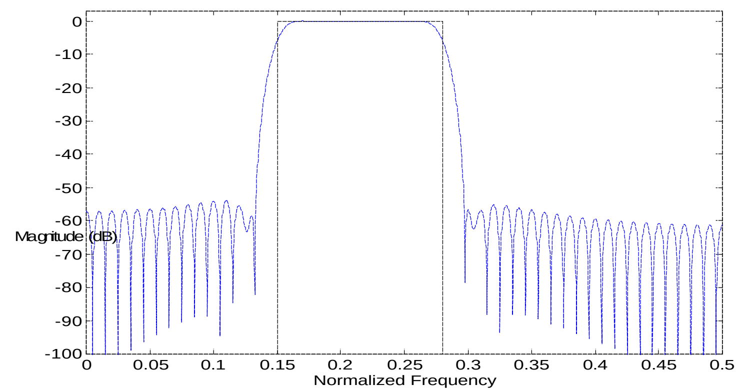
Effect of Windowing on Bandpass Filter Example

- Magnitude responses of bandpass filters with length 101 for different window functions (band edges at 0.15*Fs* and 0.28*Fs*)

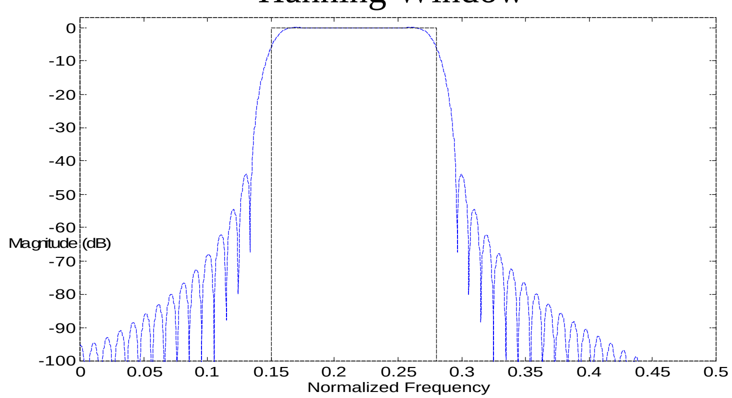
Rectangular Window



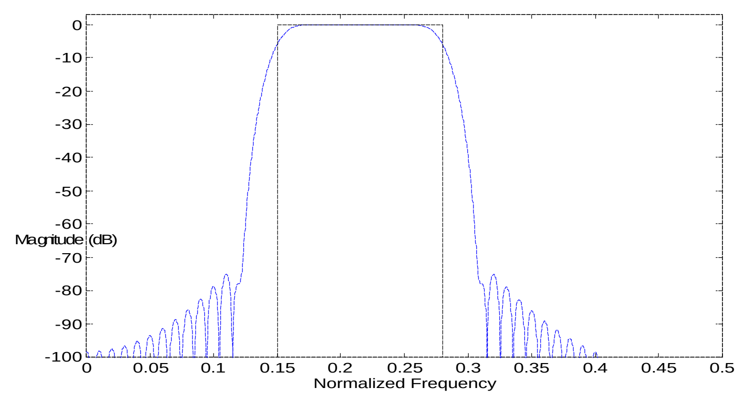
Hamming Window



Hanning Window



Blackman Window



Design Example Using the Window Method

Design a lowpass filter with passband from DC to 0.15Fs, at least 50dB attenuation above 0.2Fs, and passband ripple of less than 0.1dB.

- Any of the windows (except rectangular) will meet the passband ripple spec, but only the Hamming or Blackman will meet the stopband spec. Let’s pick the Hamming window.
- The transition band is 0.05Fs wide (ie $\Delta\omega = 0.1\pi$), so

$$\frac{3.32\pi}{M} \geq 0.1\pi \qquad \text{giving} \quad M \geq 33.2$$

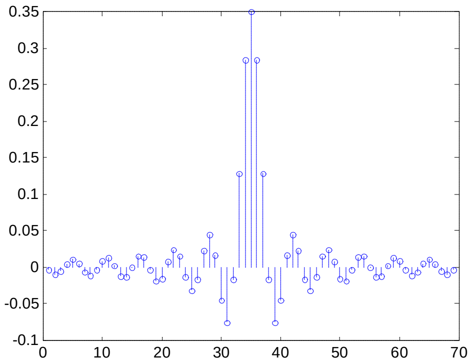
- We’ll pick a filter length of 69, giving *M* = 34.
- Next compute the ideal filter coefficients and the window coefficients, where

$$h[n] = \begin{cases} 2f_c & n = 0 \\ 2f_c \cdot \dfrac{\sin[2\pi \cdot f_c \cdot n]}{2\pi \cdot f_c \cdot n} & n \neq 0 \end{cases} \qquad w[n] = 0.54 + 0.46\cos\left(\frac{2\pi \cdot n}{2M+1}\right) \quad -M \leq n \leq M$$

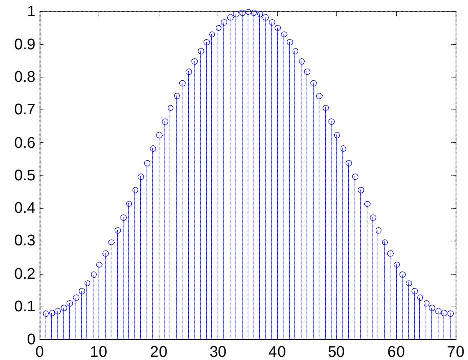
In this example $fc = \dfrac{0.15 + 0.2}{2} = 0.175$

Design Example Using the Window Method (cont)

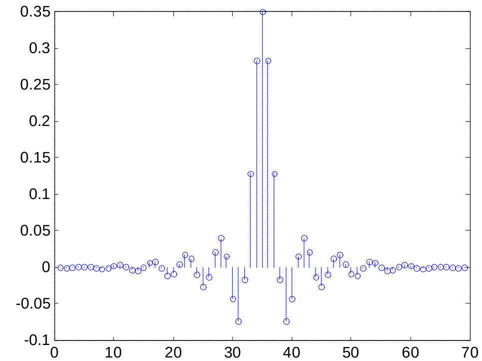
Ideal Impulse Response



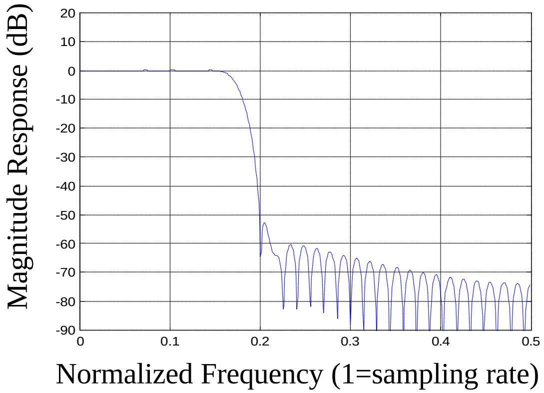
Window Function



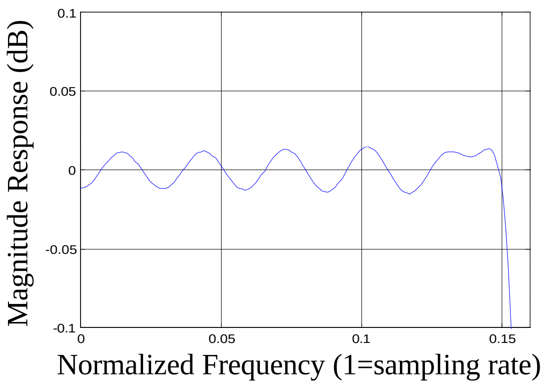
Windowed Impulse Response



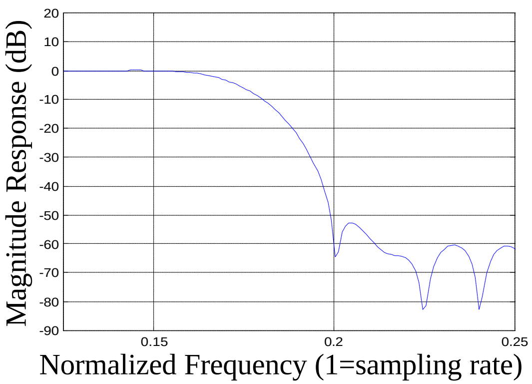
Magnitude Response



Passband



Transition Region



Optimal Design Methods for FIR Filters

- Design methods discussed so far generate filters that are sub-optimal because
  - the resulting passband and stopband ripple amplitudes are the same.
  - the passband and stopband ripple amplitudes are not constant, but decay as we move away from the discontinuities.
- The length of the filter to meet a given spec can be reduced if
  - we allow different passband and stopband ripple amplitudes.
  - we make the ripple magnitude constant in the passband and stopband.
- The most commonly used algorithm is the Parks-McClellan algorithm.

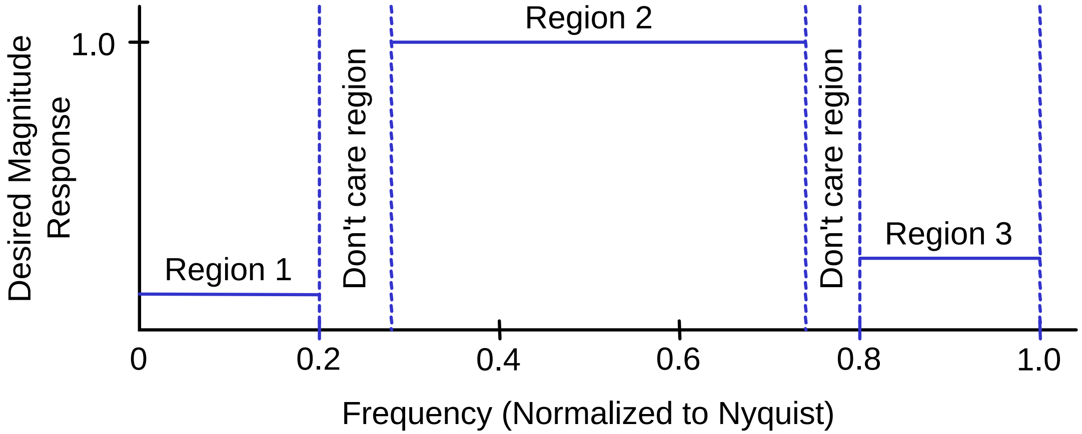
Parks-McClellan Algorithm

- The objective is to minimize the maximum error across the filter bands.
- The algorithm makes use of the *Remez Exchange* optimization.
- The algorithm is implemented in *Matlab* with the functions *remezord* and *remez*.

Design Approach

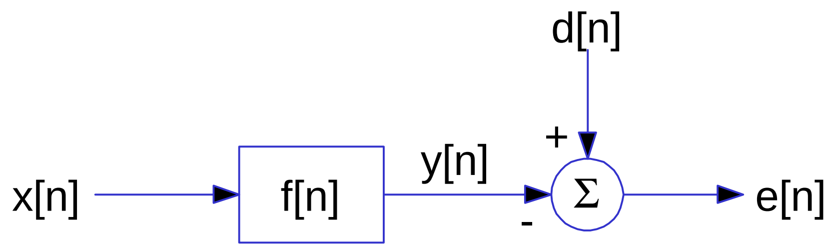
- Separate normalized frequency-space into regions that define the desired response. There should be a 'don't care' region between each 'do care' region.
- Specify a weighting factor for each region.
- Use *Matlab* to estimate the filter order, and then to compute the impulse response.

Ex.



Optimal Least-Squares Filter Design

- Consider a situation where a signal *x[n]* is to be filtered in such a way that the output sequence is as close as possible to a desired signal *d[n]*



$$e[n] = d[n] - \sum_k f[k] \cdot x[n-k]$$

- The least-squares solution involves taking the derivative of the mean-squared error with respect to each coefficient and setting the result to zero.
- The result is a set of *Normal Equations* that can be solved to find the optimum FIR filter coefficients from the input auto-correlation and input-demand cross-correlation functions.

$$\sum_j r_{xx}(j-i) \cdot f(j) = r_{dx}(j)$$

where $r_{xx}(j-i) = E\{x[n-j] \cdot x[n-i]\}$ and $r_{dx}(j) = E\{d[n] \cdot x[n-j]\}$

AVERAGING AS A FILTER

Simple Lowpass FIR Digital Filter

- The simplest FIR filter is a 2-point moving average, with transfer function

$$H_{lp}(z) = \frac{Y(z)}{X(z)} = \frac{1}{2}(1 + z^{-1})$$

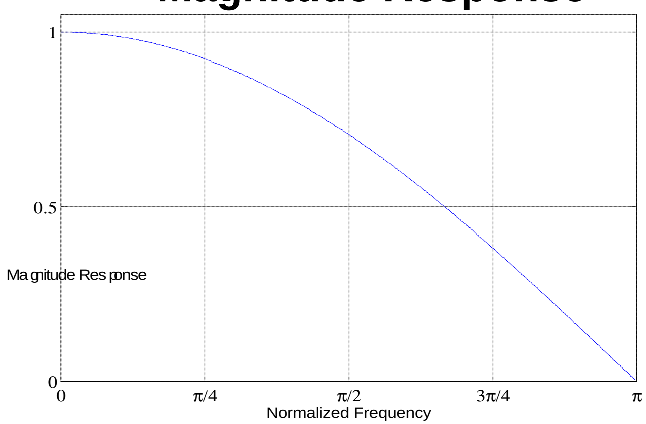
The difference equation is

$$y[n] = 0.5 \cdot \left(x[n] + x[n-1]\right)$$

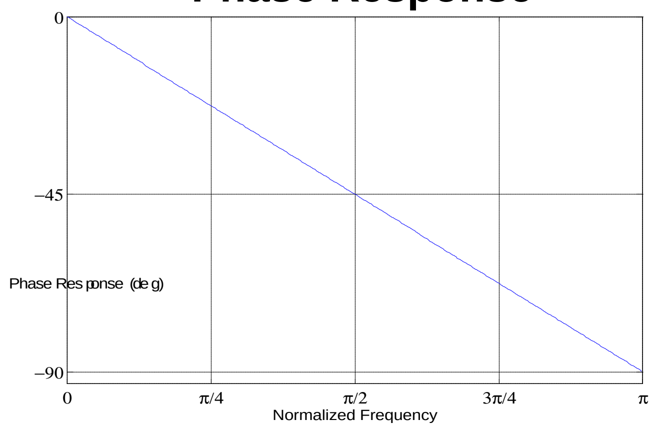
Its frequency response is given by

$$H_{lp}(e^{jw}) = \frac{1}{2}(1 + e^{-jw}) = \frac{1}{2}e^{-jw/2}(e^{jw/2} + e^{-jw/2}) = e^{-jw/2}\cos\frac{w}{2}$$

Magnitude Response



Phase Response



4-Point FIR Averager

- A 4-point moving average, has the transfer function

$$H_{lp}(z)=\frac{Y(z)}{X(z)}=\frac{1}{4}(1+z^{-1}+z^{-2}+z^{-3})$$

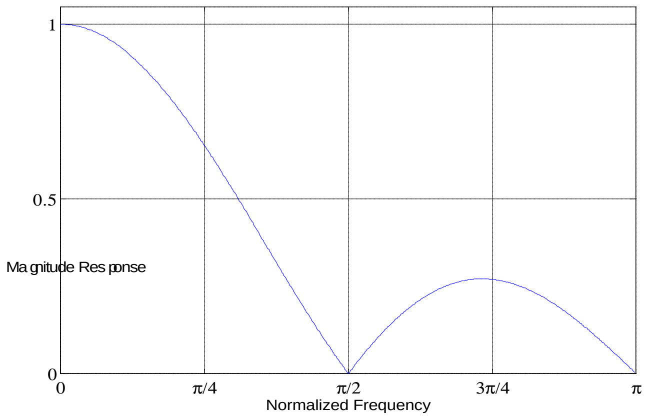
The difference equation is

$$y[n]=0.25\cdot\left(x[n]+x[n-1]+x[n-2]+x[n-3]\right)$$

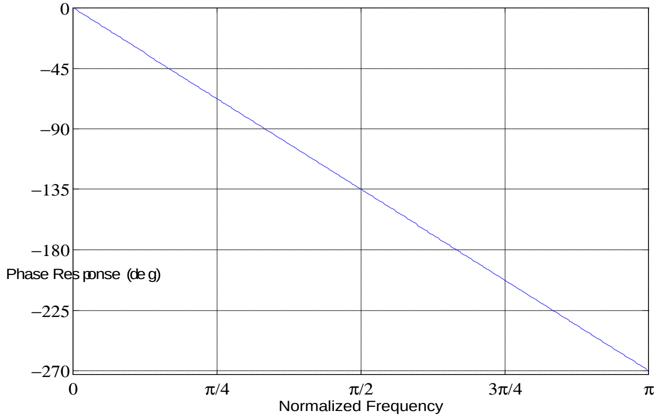
Its frequency response is given
by

$$H_{lp}(e^{jw})=e^{-j3w/2}\cdot\left[\cos w+\cos\frac{w}{2}\right]$$

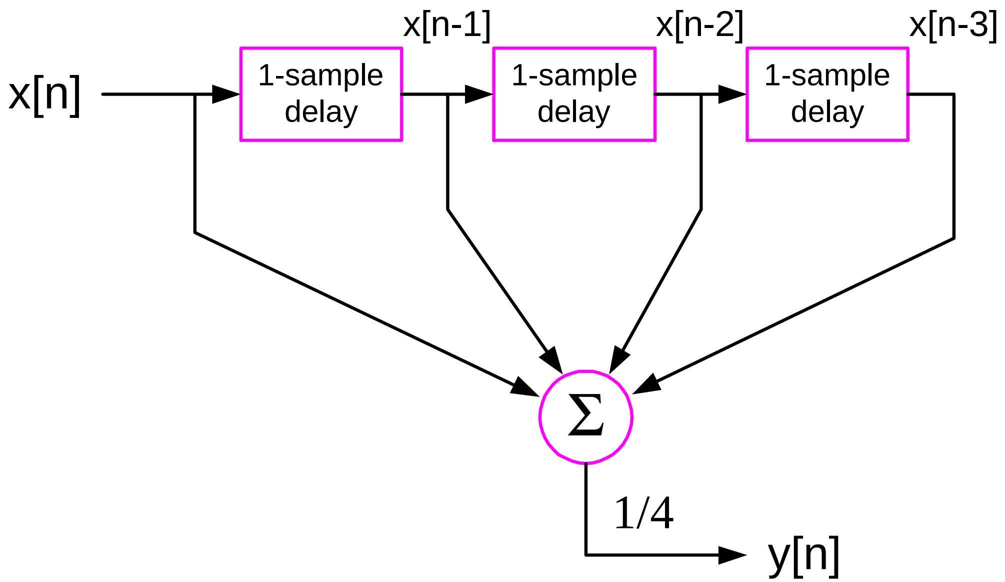
Magnitude Response



Phase Response



Averager Block Diagram (DSP Viewpoint)



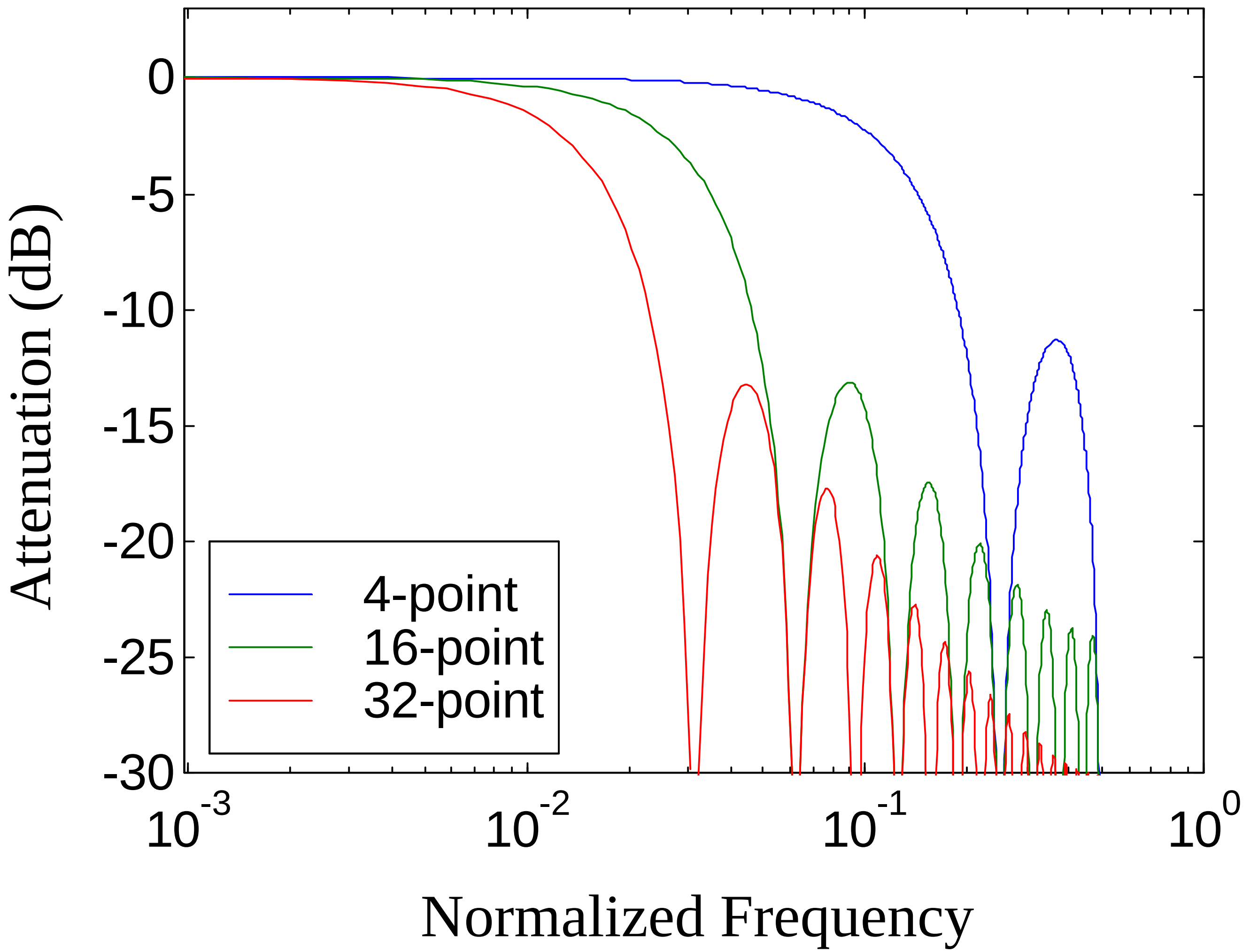
- This can be described with the following *difference equation*

$$y[n] = 0.25 \cdot \left(x[n] + x[n-1] + x[n-2] + x[n-3]\right)$$

- Or with the following z-transform transfer function

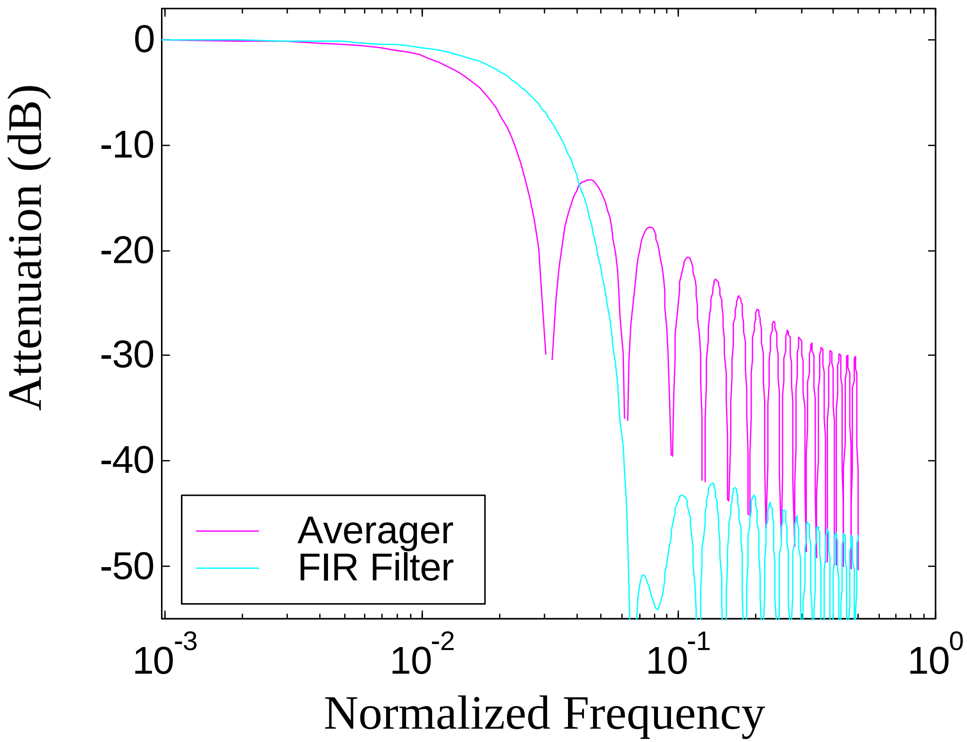
$$H_{lp}(z) = \frac{Y(z)}{X(z)} = \frac{1}{4}(1 + z^{-1} + z^{-2} + z^{-3})$$

Averagers with Different Number of Points

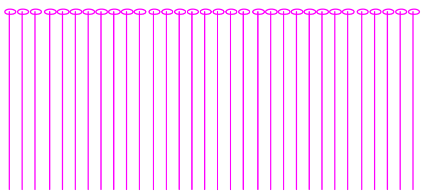


32-Tap Averager vs 32-Tap FIR Filter

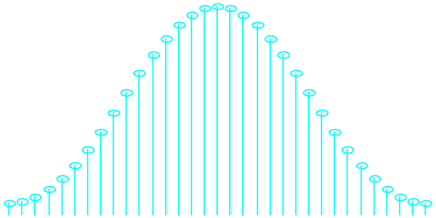
- A boxcar averager is simple to implement, but does not provide the optimum level of filtering



Averager Coefficients



FIR Filter Coefficients



Using averaging to get more effective resolution

- Single-sample (turn-by-turn) resolution of APS bpms is nominally 12-bits.
- Residual noise in the analog front-end provides an opportunity to get more resolution by averaging data samples
  - Assuming Gaussian noise, we improve the resolution by a factor 2 (one additional bit) by averaging four samples.
  - The APS bpm processing system uses a 1024-sample boxcar averager to improve the resolution by a factor 32, giving effectively 17-bit resolution.
  - In principle we can increase the resolution ad infinitum, provided we are willing to wait long enough to collect the requisite number of samples.

When does this breakdown?

- Averaging will always work when dealing with Gaussian noise, but at some point, other non-Gaussian processes start to dominate, limiting the performance
  - Front-end amplifier non-linearity.
  - Digitizer quantization errors (integral and differential non-linearity).
  - Word-length effects in the digital processing circuits.
  - Drift.
- Usually digitizers with 12-bit performance do not have 17-bit systematics.

INFINITE IMPULSE REPSONSE (IIR) DIGITAL FILTERS

IIR Digital Filter Design Methods

- Generate digital filter from analog prototype
  - generate lowpass normalized analog prototype filter.
  - convert lowpass prototype to other form if necessary (eg highpass, bandpass).
  - convert analog filter to digital domain
    - impulse invariance.
    - bilinear transform.
- Generate digital filter directly in digital domain
  - least squares design in frequency domain.
  - least squares fitting of desired discrete-time impulse response.

Simple Lowpass IIR Digital Filter

- A first-order lowpass IIR digital filter has the transfer function

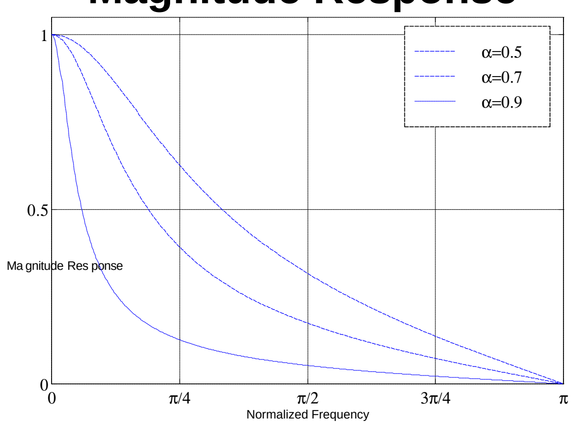
$$H_{lp}(z) = \frac{Y(z)}{X(z)} = \frac{1-a}{2}\,\frac{1+z^{-1}}{1-a\cdot z^{-1}} \qquad |a|<1$$

The difference equation is

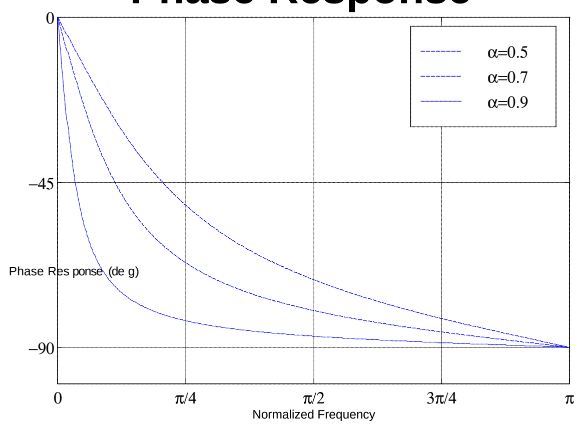
$$y[n] = \left(\frac{1-a}{2}\right)\cdot\left(x[n]+x[n-1]\right)+a\cdot y[n-1]$$

- This is the discrete-time equivalent of an electronic R-C circuit

Magnitude Response



Phase Response



IIR Digital Filter Design by Impulse Invariance Method

- The idea is to design a digital filter whose impulse response is identical to the sampled version of the impulse response of the analog filter prototype.
- Given the Laplace transfer function of an analog prototype filter $H_a(s)$, then the impulse response is given by

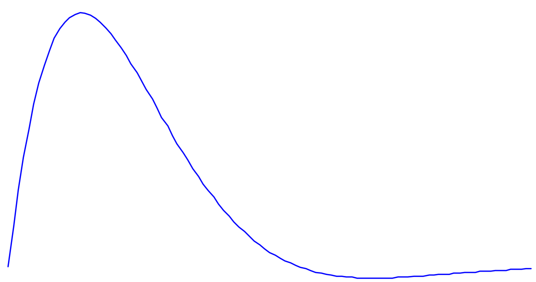
$$h_a(t) = L^{-1}\{H_a(s)\}$$

- The impulse response of the digital filter is $h_a(t)$ sampled at periodic intervals T

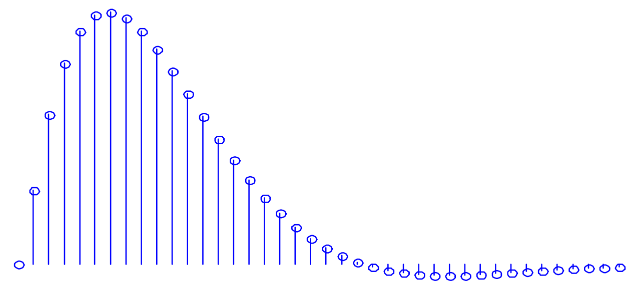
$$g[n] = h_a(nT) \qquad n = 0,1,2,3\ldots$$

- And the z-transform of the digital filter is given by

$$G(z) = Z\{g[n]\} = Z\{h_a(nT)\}$$



$$h_a(t) = L^{-1}\{H_a(s)\}$$



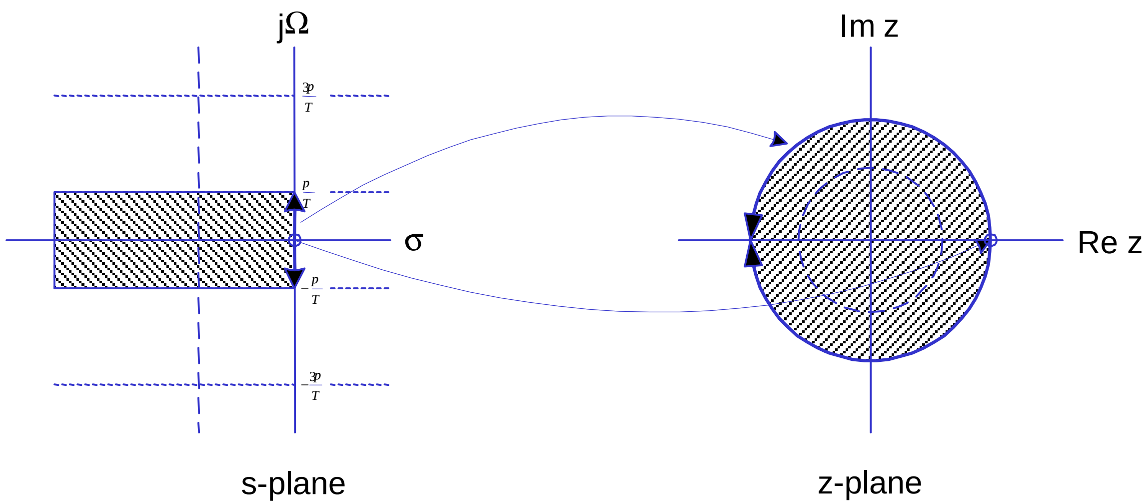
$$g[n] = h_a(nT)$$

Impulse-Invariance Mapping

- Mapping of the s-plane poles and zeros to the z-plane is achieved by the transformation

$$z = e^{sT}$$

For s = σ+jΩ, we get $$z = e^{sT} = e^{(s + j\Omega)T} = e^{sT} e^{j\Omega T}$$



- The entire strip on the s-plane between -π/2 and +π/2 is mapped into the unit circle of the z-plane.
- Because of the periodicity of the mapping, the strip on the s-plane between π/2 and 3π/2 (and all other similar strips) are also mapped into the unit circle of the z-plane.

Using the Impulse-Invariance Mapping

- Consider a simple 1-pole (stable) analog filter described by the Laplace transform

$$H(s) = \frac{A}{s + a}$$

- The continuous-time impulse response is given by

$$h(t) = Ae^{-a \cdot t}$$

- The discrete-time impulse response is obtained by sampling the *h(t)* at time intervals *T*

$$g[n] = h(nT) = Ae^{-a \cdot n \cdot T} = A\left(e^{-a \cdot T}\right)^{n}$$

- The closed-form expression for the z-transform of *g[n]* is therefore

$$G(z) = \frac{A}{1 + e^{-a \cdot T} z^{-1}}$$

Impulse Invariance Mapping of 1st and 2nd Order Poles

- So, to generate the z-transform from the Laplace transform,

we replace $\dfrac{A}{s+a}$ with $\dfrac{A}{1+e^{-a \cdot T} z^{-1}}$

- There are two forms of the second-order transfer functions, and without proof, are mapped as follows

$$H_1(s) = \frac{l}{(s+b)^2 + l^2} \rightarrow G_1[z] = \frac{ze^{-bT} \sin lT}{z^2 - 2ze^{-bT} \cos lT + e^{-2bT}}$$

$$H_2(s) = \frac{s+b}{(s+b)^2 + l^2} \rightarrow G_2[z] = \frac{z^2 - ze^{-bT} \cos lT}{z^2 - 2ze^{-bT} \cos lT + e^{-2bT}}$$

Impulse-Invariance Numerical Example

- Consider the following 2-pole filter that is to be converted to the discrete-domain at a sample rate of 20Hz.

$$H(s) = \frac{1}{s^2 + \sqrt{2}s + 1} = \frac{\sqrt{2}(1/\sqrt{2})}{(s + \sqrt{2}/2)^2 + 1/2}$$

- We will use the first form of the 2nd-order mapping,

$$H_1(s) = \frac{l}{(s+b)^2 + l^2} \rightarrow G_1[z] = \frac{ze^{-bT} \sin lT}{z^2 - 2ze^{-bT} \cos lT + e^{-2bT}}$$

So that

$$G[z] = \sqrt{2} \cdot \frac{ze^{-bT} \sin lT}{z^2 - 2ze^{-bT} \cos lT + e^{-2bT}}$$

where

$$l = 1/\sqrt{2}$$
$$b = 1/\sqrt{2}$$
$$T = 1/20$$

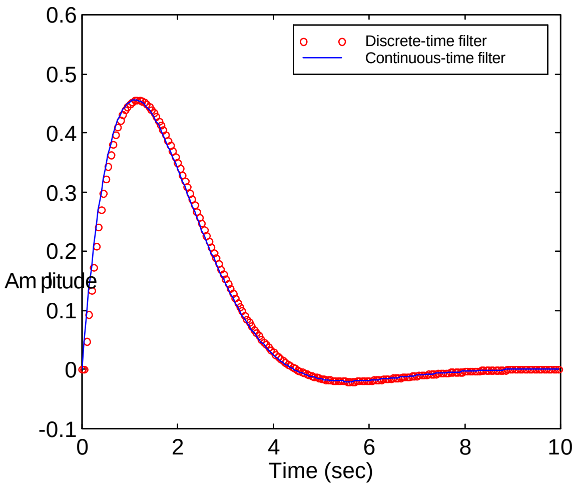
Giving

$$G_{ii}[z] = \frac{0.06824z^2}{z^2 - 1.9293z + 0.9317}$$

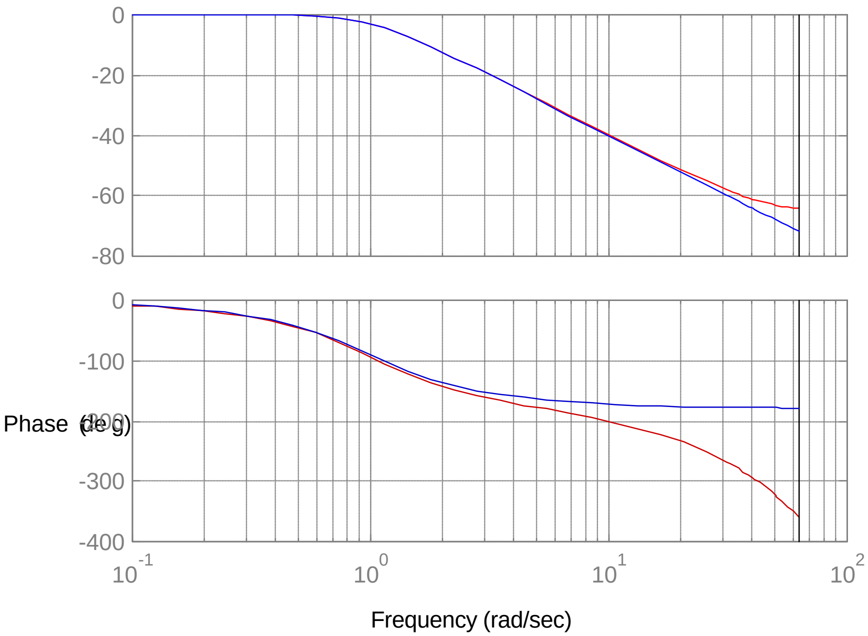
Impulse-Invariance Numerical Example (cont)

- Comparisons of the original continuous-time and the discrete-time filters are shown below.

Impulse Responses

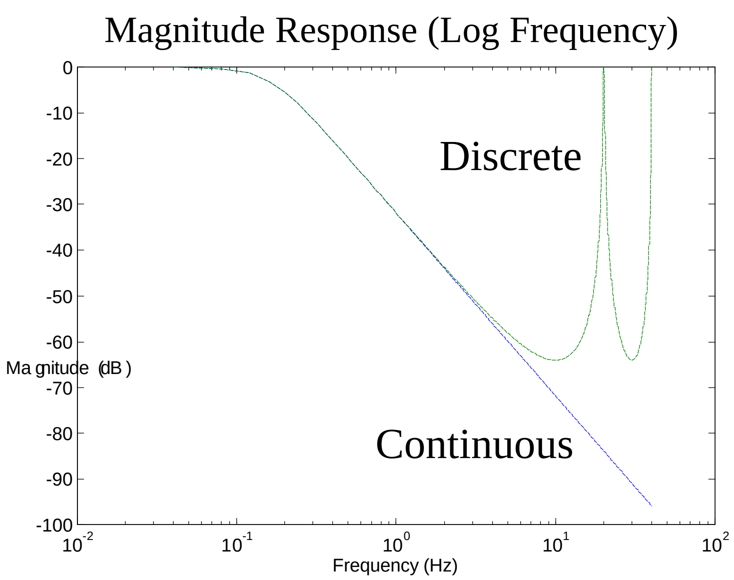


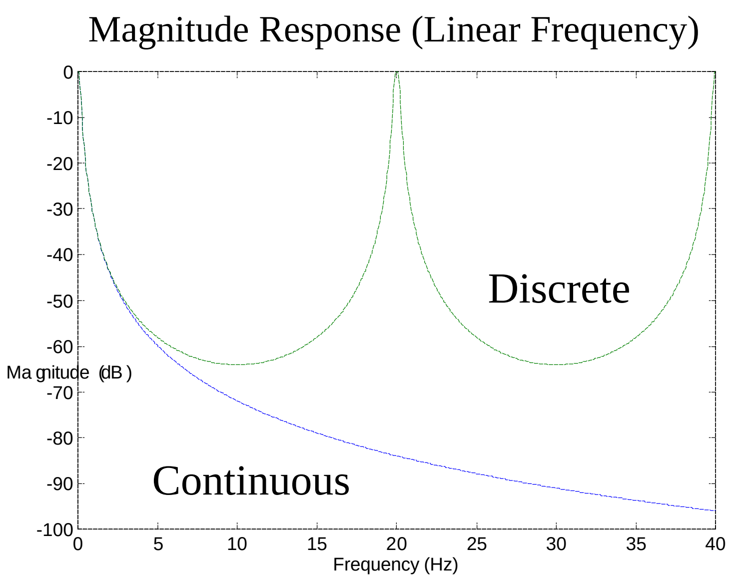
Bode Diagrams



Aliasing with the Impulse-Invariance Transformation

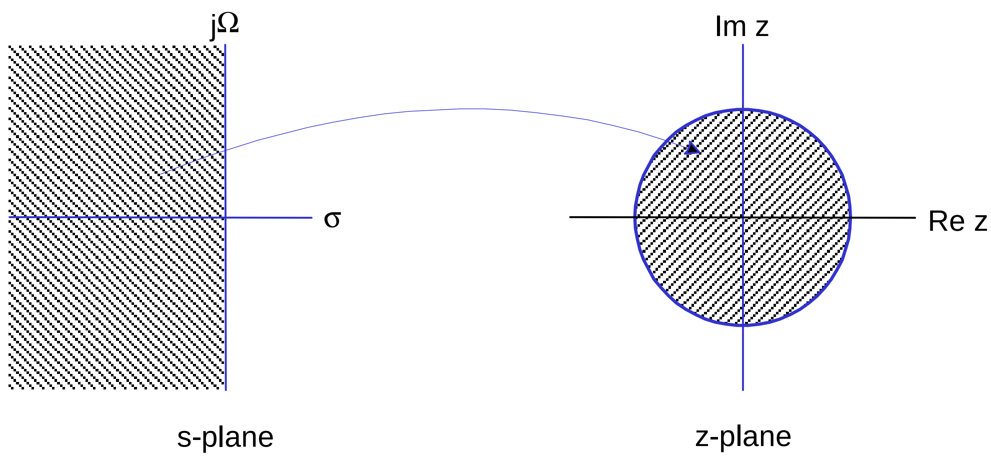
- Since the mapping is not unique, there is aliasing of the original analog frequency response above half the sampling frequency.
- The figures show the magnitude response of the same 2-pole Butterworth filter over a frequency range up to twice the sampling frequency





IIR Filter Design using the Bilinear Transformation

- Unlike the impulse-invariance transformation, the bilinear transformation maps the entire left-half of the s-plane into the unit circle.
- Because there is a one-to-one correspondence between points on the s-plane and points on the z-plane, there is no aliasing of the filter response.



- The bilinear transformation is given by

$$s \rightarrow C \cdot \left(\frac{1 - z^{-1}}{1 + z^{-1}} \right)$$

where *C* is a constant to be found

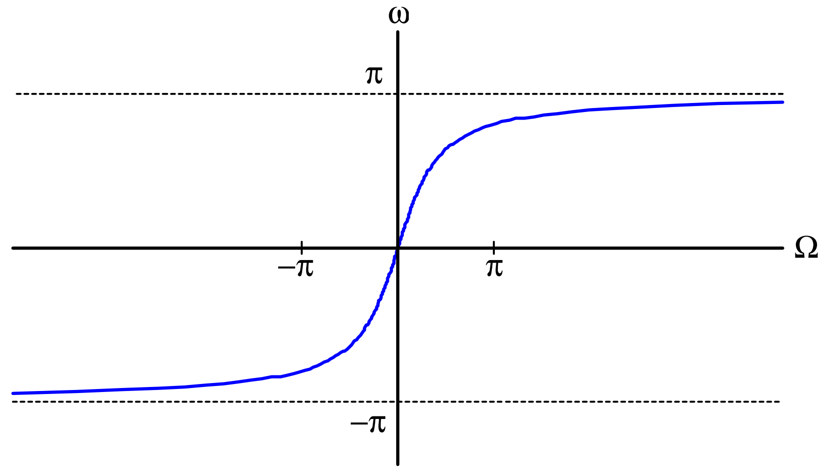
BLT Warping of Analog Frequencies to Digital Frequencies

- The mapping from analog frequency W to discrete-time frequency w is

$$\Omega = C \cdot \tan\frac{w}{2} \qquad \text{where} \quad w = \frac{2p \cdot F_C}{F_S}$$

and C is a mapping constant

- The mapping is shown graphically below



$$\Omega = 0 \rightarrow w = 0$$

$$\Omega = \infty \rightarrow w = p$$

- The mapping constant allows us to adjust the scaling so we can get exact correspondence at one additional frequency. A low frequency approximation is

$$C = \frac{2}{T} = 2F_s$$

BLT Example

Design a digital IIR filter that implements the analog lowpass filter described by the following normalized Laplace transfer function and a sampling rate of 20Hz. Use the low frequency approximation of the bilinear transformation.

$$H_{lp}(s) = \frac{1}{s^2 + \sqrt{2}s + 1}$$

- We will use the following mapping to get good low frequency approximation

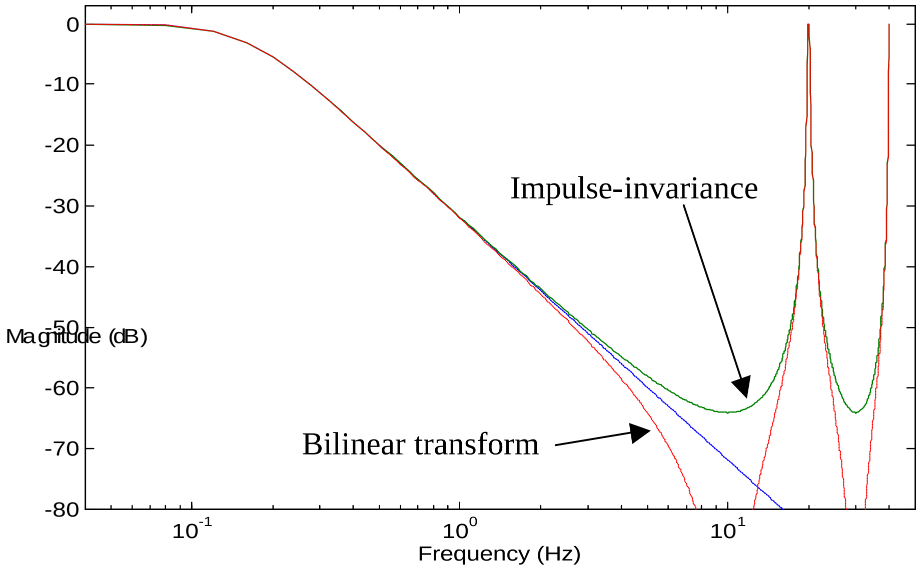
$$s \to \frac{2}{T} \cdot \left(\frac{1 - z^{-1}}{1 + z^{-1}} \right) = 40 \cdot \left(\frac{1 - z^{-1}}{1 + z^{-1}} \right)$$

- Plugging this into the Laplace transfer function gives

$$G_{blt}[z] = \frac{1}{\left(40 \cdot \frac{1 - z^{-1}}{1 + z^{-1}} \right)^2 + \sqrt{2}\left(40 \cdot \frac{1 - z^{-1}}{1 + z^{-1}} \right) + 1} = \frac{0.0006033 \cdot (1 + 2z^{-1} + z^{-2})}{1 - 1.9293z^{-1} + 0.9375z^{-2}}$$

Comparison with Impulse Invariance Method

- The magnitude responses for the original continuous-time filter and the discrete-time filters from both impulse invariance and bilinear transformation are show below



- For low frequencies, the impulse invariance method gives an exact match with the continuous-time filter.
- The bilinear transformation generates a zero (null) response at the Nyquist frequency, whereas the impulse invariance aliases the original response.
- Both discrete-time filters have alias responses about multiples of the sampling rate.

BLT Bandpass Example with Pre-warping

Design a digital bandpass filter given the following Laplace transfer function that has a passband from 100rad/s to 200rad/s. The sample rate should be 100Hz. Use the bilinear transformation such that the upper band edge matches exactly.

$$H_{bp}(s) = \frac{4s^2}{s^4 + 2.8284s^3 + 10s^2 + 8.4853s + 9}$$

- First we have to determine the value of the mapping constant C in the transformation

$$\Omega = C \cdot \tan\frac{w}{2} \rightarrow C = \Omega \cdot \cot\frac{w}{2}$$

- The value of ω is determined from the sampling rate and desired matching frequency

$$w = 2p\,\frac{F_c}{F_s} = \frac{200}{1000} = 0.2$$

- The analog frequency we want to match is 200rad/s so, we can compute *C* as follows

$$C = \Omega \cdot \cot\frac{w}{2} = 200 \cdot \cot\frac{0.2}{2} = 1993.3$$

BLT Bandpass Example with Pre-warping (cont)

- We can now apply the following mapping to our analog transfer function

$$s \rightarrow 1993.3 \cdot \frac{1 - z^{-1}}{1 + z^{-1}}$$

Plugging this mapping into the Laplace transfer function gives us the Z transform

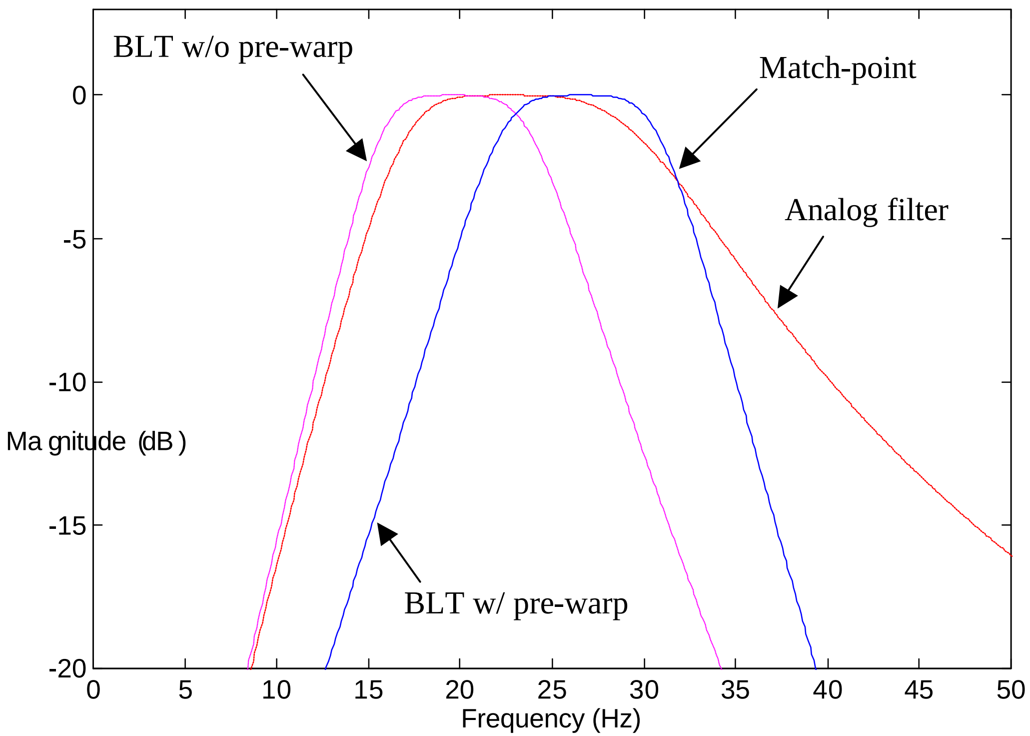
$$G_{bp}[z] = \frac{4\left(1993.3 \cdot \frac{1-z^{-1}}{1+z^{-1}}\right)^2}{\left(1993.3 \cdot \frac{1-z^{-1}}{1+z^{-1}}\right)^4 + 2.8284\left(1993.3 \cdot \frac{1-z^{-1}}{1+z^{-1}}\right)^3 + 10\left(1993.3 \cdot \frac{1-z^{-1}}{1+z^{-1}}\right)^2 + 8.4853\left(1993.3 \cdot \frac{1-z^{-1}}{1+z^{-1}}\right) + 9}$$

After a lot of manipulation, we get the discrete-time transfer function

$$G_{bp}[z] = \frac{0.07638 + 3.386 \cdot 10^{-15}\, z^{-1} - 0.1528\, z^{-2} + 7.772 \cdot 10^{-16}\, z^{-3} + 0.07638 z^{-4}}{1 + 0.2962 z^{-1} + 1.104 z^{-2} + 0.1782 z^{-3} + 0.3862}$$

BLT Bandpass Example with Pre-warping (cont)

- The resulting frequency response is plotted in the figure, together with the corresponding analog filter response and the BLT discrete-time filter response without pre-warping.



Digital PID Regulator

- The most common feedback regulator is the PID regulator, which has the Laplace transfer function

$$H_{pid}(s) = K_p + \frac{K_i}{s} + s \cdot K_d$$

Where *Kp, Ki, Kd* are the gain constants for the proportional, integral, and derivative terms, respectively.

- A digital PID can be generated from this using either BLT or impulse invariance mapping.
- In the case of the impulse invariance method, we simply use the mapping

$$s \rightarrow \frac{1}{1 - z^{-1}}$$

- This results in the discrete-time PID transfer function

$$H_{pidz}(z) = K_p + \frac{K_i}{1 - z^{-1}} + K_d \left(1 - z^{-1}\right)$$

Digital PID Regulator (cont)

- For the BLT, with a good low frequency approximation, we use the mapping

$$s \rightarrow \frac{2}{T} \cdot \left(\frac{1 - z^{-1}}{1 + z^{-1}} \right)$$

where T is the sampling interval

- This results in the discrete-time PID transfer function

$$H_{pidz}(z) = K_p + K_i \frac{2}{T} \left(\frac{1 + z^{-1}}{1 - z^{-1}} \right) + K_d \frac{2}{T} \left(\frac{1 - z^{-1}}{1 + z^{-1}} \right)$$

Comparison of FIR and IIR Filters

Characteristic	IIR Filters	FIR Filters
Filter order for given specification	Lowest	Highest
Number of multiplications	Least	Most
Memory requirements	Least	Most
Stability	Must be designed in	Guaranteed
Linear phase	Not possible	Yes if impulse response is symmetrical
Can simulate analog filters	Yes	No
Supports adaptive filtering	Yes, but non-linear solution	Yes, and linear solution
Sensitivity to coefficient quantization	Can be high – depends on realization	Generally very low
Difficulty in analyzing finite wordlength effects	More difficult	Easier